PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-260832
(43)Date of publication of application : 29.09.1998

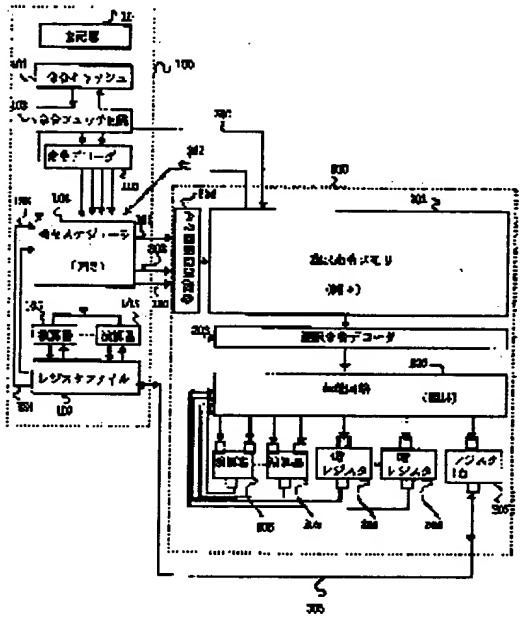
(51)Int.Cl. G06F 9/38
G06F 9/38

(21)Application number : 09-064327
(22)Date of filing : 18.03.1997

(71)Applicant : HITACHI LTD
(72)Inventor : MIKI YOSHIO
TSUSHIMA YUJI

(54) INFORMATION PROCESSOR

(57)Abstract:
PROBLEM TO BE SOLVED: To shorten the schedule time to parallelly execute an instruction for successive execution.
SOLUTION: An instruction interpretation circuit 204 in a processor 200 generates a group of interpretation instructions to branch blocks of a program and stores it in interpretation instruction memory 201 when a scalar processor 100 carries out the program. Each interpretation instruction includes the numbers of physical resources (a computing element, a register, etc.) of a transfer source and a transfer destination which are decided by an instruction scheduler 104 to each of plural instructions that are decided as parallelly executable by the scheduler 104. When the branch block is executed again later, an interpretation instruction string in the memory 201 is successively carried out. A transfer circuit 203 supplies data to an input of a computing element that is selected for an instruction when the computing element of the transfer source generates the data that is used for an operation which is requested by each interpretation instruction.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－260832

(43)公開日　平成10年(1998)９月29日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/38	３１０	Ｇ０６Ｆ　9/38	３１０Ｆ
	３７０		３７０Ｘ

審査請求　未請求　請求項の数９　ＯＬ　（全 22 頁）

(21)出願番号　特願平9－64327

(22)出願日　平成９年(1997)３月18日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　三木　良雄
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　對馬　雄次
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

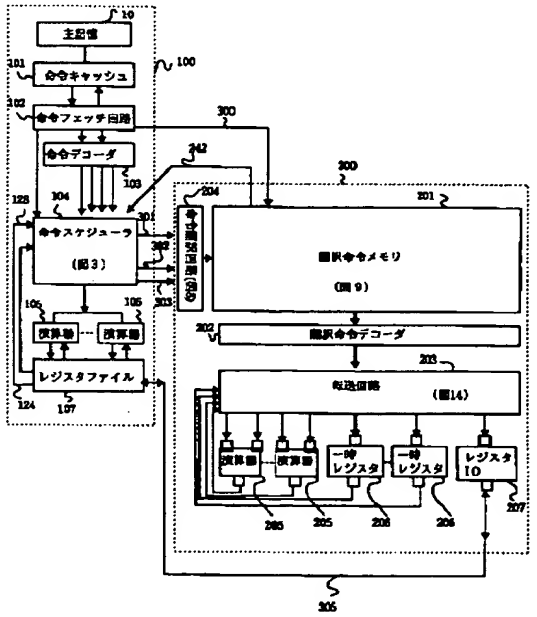
(54)【発明の名称】　情報処理装置

(57)【要約】

【課題】逐次実行用の命令の並列実行のためのスケジュール時間を短縮する。

【解決手段】スカラープロセッサ１００でプログラムが実行されたときに、そのプログラム中の分岐ブロックに対して、プロセッサ２００内の命令翻訳回路２０４が一群の翻訳命令を生成し、翻訳命令メモリ２０１に記憶する。各翻訳命令は、命令スケジューラ１０４が並列に実行可能と判断した複数の命令の各々に対して、その命令スケジューラ１０４が決定した転送元および転送先の物理資源（演算器、レジスタなど）の番号を含む。分岐ブロックが後に再実行されるときには、翻訳メモリ２０１内の翻訳命令列が順次実行される。転送回路２０３は、各翻訳命令が要求する演算に使用するデータが転送元の演算器により生成されたときに、そのデータをその命令のために選ばれた演算器の入力に供給する。

図１



1

【特許請求の範囲】

【請求項１】複数の第１種の演算器と複数の第１種のレジスタを含む複数の第１種の物理資源と、

実行すべきプログラムに含まれた複数の第１種の命令の内、並列に実行可能な複数の第１種の命令を選択し、それぞれの第１種の命令の実行に使用する複数の第１種の物理資源をそれらの選択された複数の第１種の命令に割り当て、それらの複数の第１種の命令のそれぞれに割り当てられた物理資源を使用してそれらの第１種の命令を並列に実行するように上記選択された複数の第１種の命令の実行を制御する命令スケジューラと、

上記命令スケジューラにより複数の第１種の命令が並列に実行されるごとに、該並列に実行された複数の第１種の命令の各々に対して上記命令スケジューラにより割り当てられた複数の物理資源を識別可能な少なくとも一つの第２種の命令を記憶するメモリと、

複数の第２種の演算器と複数の第２種のレジスタを含む複数の第２種の物理資源と、

上記プログラムが再度実行されたときに、そのプログラムに含まれた複数の第１種の命令に代えて、該複数の第１種の命令に対応して上記メモリに記憶された少なくとも一つの第２種の命令を実行する命令実行回路とを有し、

上記命令実行回路は、その第２種の命令に対応する上記複数の第１種の命令の各々に対して割り当てられた、上記一つの第２種の命令から識別可能な複数の物理資源に対応する複数の第２種の物理資源を使用してその第２種の命令を実行する情報処理装置。

【請求項２】上記複数の第２種の演算器は、上記複数の第１種の演算器とは別に設けられ、上記複数の第２種のレジスタは、上記複数の第１種のレジスタとは別に設けられている請求項１記載の情報処理装置。

【請求項３】上記複数の第１種の演算器が上記複数の第２種の演算器として使用され、上記複数の第１種のレジスタが上記複数の第２種のレジスタとして使用される請求項１記載の情報処理装置。

【請求項４】上記複数の第２種の演算器は、上記複数の第１種の演算器と同じ演算器を含み、上記複数の第２種のレジスタの数は、上記複数の第１種のレジスタの数と等しい請求項１記載の情報処理装置。

【請求項５】上記複数の第２種の演算器は、上記複数の第１種の演算器と異なる数の演算器を含み、上記複数の第２種のレジスタの数は、上記複数の第１種のレジスタの数とは異なる請求項１記載の情報処理装置。

【請求項６】複数の演算器と複数のレジスタを含む複数の物理資源と、

それぞれ一組の並列に実行可能な複数の第１種の命令に対応する複数の第２種の命令を記憶するメモリと、

上記翻訳メモリに記憶された複数の第２種の命令を順次実行する命令実行回路とを有し、

2

上記複数の第２種の命令は、それぞれの第２種の命令は対応する複数の第１種の命令が要求する演算を実行するために使用する複数の物理資源を識別可能に上記メモリに記憶され、

上記命令実行回路は、各第２種の命令に対応する複数の第１種の命令の各々に対して割り当てられた、上記第２種の命令から識別可能な複数の物理資源を使用してその第２種の命令を実行する情報処理装置。

【請求項７】複数の演算器と、

複数のレジスタと，命令によって使用される演算器およびレジスタを決定し，命令実行の入力となるレジスタ内データの更新を監視することによって，実行可能な状態となった命令を順次演算器へ投入する命令スケジューラと演算器へ投入された上記命令の命令語格納アドレスと使用される演算器およびレジスタ情報を記憶するメモリと、

既に上記メモリに格納されている命令語格納アドレスの命令を再度実行する際に，上記メモリに格納されている使用する演算器とレジスタに関する情報に基づきその命令を実行する命令実行回路とを有する情報処理装置。

【請求項８】複数の演算器と、

複数のレジスタと，命令によって使用される演算器およびレジスタを決定し、命令実行の入力となるレジスタ内データの更新とさらに命令実行の入力となるレジスタ内データを生成する演算器を監視することによって、命令実行に必要なデータの演算器からレジスタまたは演算器から演算器への転送経路を決定する転送回路と、

演算器へ投入された上記命令の命令語格納アドレスと命令実行に必要なデータの転送経路情報を記憶するメモリとを有し、

既に上記メモリに格納されている命令語格納アドレスの命令を再度実行する際には、上記転送回路は、上記メモリに格納されている転送経路情報に基づき演算器とレジスタの間のデータ転送経路を切り替えて上記命令を実行する回路を有する情報処理装置。

【請求項９】複数の演算器と、

複数のレジスタと、

上記複数の演算器および複数のレジスタ間のデータ転送を可能とする出力線と入力線との間の複数の交点位置に設けられた、実行待ちの命令を待機させるための複数の命令待機回路と、

命令実行に必要なデータを生成する演算器または命令実行に必要なデータを保持しているレジスタと演算が実行される演算器との識別情報をデータ転送情報として持つ実行待ちの命令を、上記複数の命令待機回路の内、その命令の実行にためにデータ転送が必要となる演算器またはレジスタの入出力線の交点位置に設けられた一つの命令待機回路に投入する回路とを有する情報処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術の分野】本発明は、逐次実行用のプログラムの命令を並列に実行するスーパースカラー方式の情報処理装置に関する。

【０００２】

【従来の技術】命令により制御される情報処理装置の処理速度を向上させるために、命令列が潜在的に有する並列実行性を利用する方法がある。例えば、マイク・ジョンソン著“スーパースカラー・プロセッサ”日経ＢＰ出版センタ、１９９４年、第１０４頁から第１１８頁および第１２５頁から第１４４頁（以下、文献１と呼ぶ）に示されているようなスーパースカラー方式では、複数の演算器が１つのプロセッサ内に設けられ、複数の命令を異なるの演算器を使って並列に実行する。並列実行可能な命令は、命令実行ステージ以前のパイプラインステージ、例えば命令デコードステージにおいて、複数の命令が利用するハードウエア資源の競合や、命令実行に伴うデータの依存関係を調べることによって抽出される。

【０００３】文献１の１２５ページ以降に記載されているように、アウトオブオーダ方式も可能である。この方法では、並列実行可能な命令を検索する範囲を広げ、並列実行可能な命令を元の命令順序とは異なる順序で実行する。

【０００４】アウトオブオーダ実行を実現す方法として、文献１の１０９頁に記載されているＴｏｍａｓｕｌｏアルゴリズムが知られている。Ｔｏｍａｓｕｌｏアルゴリズムでは、ある命令のデコードが終了し、その命令が利用する演算器や必要とする入力データが解読された後に、その命令がリザベーションステーションと呼ばれる一種の命令バッファに格納される。リザベーションステーションは実行待ちの命令格納場所として用いられ、演算器で実行終了した命令は、その終了をリザベーションステーション内の全ての命令に告知する。これにより、待ち状態にある命令は自分自身の実行可能性を調べることができ、実行可能となった命令から順に演算器へ送られる。その結果として元の命令列順序とは異なった順序で命令を実行可能となる。２命令間の細かな依存関係の利用法としては、Ｈｅｎｎｅｓｙａｎｄ Ｐａｔｔｅｒｓｏｎ，"Ｃｏｍｐｕｔｅｒ ＡｒｃｈｉｔｅｃｔｕｒｅＡ Ｑｕａｎｔｉｔａｔｉｖｅ Ａｐｐｒｏａｃｈ（ｓｅｃｏｎｄ ｅｄｉｔｉｏｎ）", Ｍｏｒｇａｎ Ｋａｕｆｍａｎｎ Ｐｕｂｌｉｓｈｅｒｓ，Ｉｎｃ．１９９５，ＰＰ．１４７（以下、文献２と呼ぶ）に記載されているように、データフォワーディングが知られている。この方法では、或るレジスタの値を入力データとする命令が存在した時に、そのレジスタの値が確定した後に命令の実行を開始するのではなく、そのレジスタの値を更新しようとしている演算器から直接に値をその命令に転送し、レジスタの値が確定する前にその命令の実行を開始する。

【０００５】このように、命令の実行に先立ってそれらの命令間の関係を認識する方式は、命令が利用するハードウエア資源を具体的に決定すること、必要なデータが準備でき次第命令実行を開始することを制御方式として実現している。しかし、この制御方式を実現するための組み合わせ論理は大規模になる傾向が強く、ややもすれば高速実行のために設けた回路によって、全体の実行時間が消費されるという事態に陥る可能性もある。このような事態を回避するための一案として、ＶＬＩＷ（Ｖｅｒｙ Ｌｏｎｇ Ｉｎｓｔｒｕｃｔｉｏｎ Ｗｏｒｄ）（文献２、第２７８頁－第２８９頁）が提案されている。ＶＬＩＷではコンパイラによって並列実行可能な複数の命令を予め抽出し、それらの命令を一つの命令語としてまとめておく。これにより、上記の認識に必要な回路規模を抑制することが可能となる。しかしながら、命令語が変わるために全てのプログラムをコンパイルし直す必要がある。

【０００６】

【発明が解決しようとする課題】上述のように、ＶＬＩＷは実行可能形式のバイナリーファイル資産を有効利用できない。さらに、スーパーコンピュータや信号処理専用プロセッサが対象とする、ループを多く使用する数値演算プログラムなどでは、コンパイルによるスケジューリングが可能であるが、これらはごく一部の特殊例に過ぎない。一般のプログラムでは、頻繁な分岐命令やメモリアクセスなど命令実行時間の動的変化が存在するため、コンパイラで静的にハードウエア利用順序を最適化するには限度がある。そこで、汎用プロセッサとしてはスーパースカラー方式のように、ハードウエアを用いた命令並列性の抽出および命令実行順序の決定、つまりスケジューリングが重要である。ところが、ハードウエアを用いたスケジューリングが必要とする論理量が膨大なことから、このスケジューリング部分の動作速度がプロセッサ全体の動作速度を律速してしまうという問題がある。

【０００７】また、リザベーションステーションでは、明らかにまだ実行不可能な命令に対しても、実行可能性のチェックが無駄に実施される。さらに演算器の数より多くの命令が実行可能となった場合には、実行すべき命令を選抜するなどの処理も必要である。この問題は、演算器における命令実行の前段階での時間浪費につながる。

【０００８】本発明の目的は、より少ないオーバヘッドで複数の命令を並列に実行できる情報処理装置を提供するにある。

【０００９】本発明のより具体的な目的は、命令のスケジューリングによる命令の実行速度の低下を低減できる情報処理装置を提供することにある。

【００１０】本発明の他のより具体的な目的は、命令が必要とするデータが準備でき次第、より少ない遅延でもってその命令の実行を開始できる情報処理装置を提供す

ることにある。

【００１１】

【課題を解決するための手段】上記目的を達成するために、本発明による情報処理装置には、複数の第１種の演算器と複数の第１種のレジスタを含む複数の第１種の物理資源と、実行すべきプログラムに含まれた複数の第１種の命令の内、並列に実行可能な複数の第１種の命令を選択し、それぞれの第１種の命令の実行に使用する複数の第１種の物理資源をそれらの選択された複数の第１種の命令に割り当て、それらの複数の第１種の命令のそれぞれに割り当てられた物理資源を使用してそれらの第１種の命令を並列に実行するように上記選択された複数の第１種の命令の実行を制御する命令スケジューラとが設けられ、これらを用いてスーパースカラーモードで上記プログラムの第１種の命令が実行される。　本発明では、上記命令スケジューラにより複数の第１種の命令が並列に実行されるごとに、該並列に実行された複数の第１種の命令の各々に対して上記命令スケジューラにより割り当てられた複数の物理資源を識別可能な少なくとも一つの第２種の命令を記憶するメモリと、複数の第２種の演算器と複数の第２種のレジスタを含む複数の第２種の物理資源と、上記プログラムが再度実行されたときに、そのプログラムに含まれた複数の第１種の命令に代えて、該複数の第１種の命令に対応して上記メモリに記憶された少なくとも一つの第２種の命令を実行する命令実行回路とが設けられ、上記命令実行回路は、その第２種の命令に対応する上記複数の第１種の命令の各々に対して割り当てられた、上記一つの第２種の命令から識別可能な複数の物理資源に対応する複数の第２種の物理資源を使用してその第２種の命令を実行する。

【００１２】この結果、上記プログラム内の第１種の命令列を再度実行するときには、このメモリに記憶してある第２種の命令を実行することで、元の第１種の命令の実行時の命令の実行順序および使用する演算器等の物理資源の情報を利用することになる。この結果、命令スケジューラにより再度並列実行可能性の認識をする必要がなく、第２種の命令列はより高速に実行できる。

【００１３】さらには、上記メモリおよび命令実行回路の動作クロックを、上記命令スケジューラのそれよりも高めることにより、第２種の命令列の実行速度を速めることも可能である。このためには、上記複数の第２種の物理資源は、上記複数の第１種の物理資源とは別に設けることが望ましい。しかし、これらの物理資源を共通の物理資源により実現することも可能である。

【００１４】本発明のより具体的な態様では、各第２種の命令は、使用する物理資源と命令実行に必要なデータを生成する物理資源の情報を持ち、それらの情報を位置情報として対応づけられた第２種の命令の転送回路内に、各第２種の命令の実行を待ちあわせる命令待ち合わせ回路が設けられる。これにより、リザーベーションステーションで起きていた、明らかに実行が不可能な命令に対する実行可能性チェックや、実行直前での命令選抜に必要な論理や時間の削減が可能となる。

【００１５】

【発明の実施の形態】以下、本発明に係る情報処理装置を、図面に示したいくつかの実施の形態を参照してさらに詳細に説明する。なお、以下においては、同じ参照番号は同じものもしくは類似のものを表わすものとする。また、第２の実施の形態以降では、第１の実施の形態との相違点を主に説明する。

【００１６】＜発明の実施の形態＞

（１）装置構成

図１において、情報処理装置は、二つのプロセッサ１００と２００により構成される。１００は、逐次実行用に作成されたプログラムから命令の並列実行可能性を抽出し，個々のスカラ命令を並列に実行可能なスーパースカラープロセッサである。スーパースカラープロセッサ１００は、主記憶１０に接続された命令キャッシュ１０１と、命令フェッチ回路１０２、命令デコーダ１０３、命令スケジューラ１０４、複数の演算器１０５およびレジスタファイル１０７を有する。２００は、スーパースカラープロセッサ１００により実行されたプログラムが再度実行されるときに、そのプログラム内の命令をスーパースカラープロセッサ１００よりも高速に実行するためのプロセッサである。プロセッサ２００は、スーパースカラープロセッサ１００より速いマシーンクロックで動作するように構成されている。スーパースカラープロセッサ１００において上記プログラムが最初に実行されたときに、命令スケジューラ１０４によるスケジュールの結果として、並列に実行される命令のアドレス３０１、各命令に割り当てられた、演算器およびレジスタなどの物理資源の番号、具体的には、その命令の実行結果データを転送する転送先物理資源の番号３０２およびその命令の実行に使用するデータを供給する転送元物理資源番号３０３が供給される。このプロセッサ２００では、命令翻訳回路２０４が、これらのスケジュール結果を反映した、それらの命令と等価な処理を指定する命令（以下これを翻訳命令と呼ぶ）を生成し、生成された翻訳命令列を翻訳命令メモリ２０１に記憶する。

【００１７】プロセッサ２００には、ここに記憶された翻訳命令を実行するための複数の演算器２０５と、これらの演算器が使用するデータあるいはこれらの演算器が生成した演算結果データを保持するための複数の一時レジスタ２０６と、これらの一時レジスタ２０６とレジスタファイル１０７との間でデータを交換するために使用されるレジスタＩＯ２０７が設けられている。これらの演算器２０５の数は、スーパースカラープロセッサ１００内の演算器１０５のそれと同じであり、一時レジスタ２０６の数もレジスタファイル１０７に含まれた、命令で指定可能なレジスタの数と同じである。もっとも、翻

訳命令で使用される物理資源数に依存してより少ない演算器２０５，一時レジスタ２０６で構成することも可能である。

【００１８】翻訳命令メモリ２０１に記憶された翻訳命令の各々は、複数のスカラ命令に関する命令情報を含み、デコーダ２０２は、翻訳命令メモリ２０１から読み出された翻訳命令を解読し、転送回路２０３は、デコーダ２０２により与えられる命令解読情報が指定する複数の演算の各々が必要とするデータが上記複数の演算器２０５のいずれかから供給されるのに同期して、その演算を開始する。複数の演算器２０５は、複数のスカラ命令が要求する演算を並列に実行するとともに、スーパースカラープロセッサ１００内の複数の演算器１０５よりも高速に演算を実行可能に構成されている。

【００１９】転送回路２０３は、命令スケジューラ１０４と同じく実行すべき命令が実行可能になるのを待ち、実行可能になった時点でその命令を、いずれかの演算器に分配することを基本的な機能としている。しかし、転送回路２０３は、命令スケジューラ１０４と異なり、後述するように、どの先行命令の実行終了を待機すべきかについては転送経路中の位置として既にデコードされた状態となる。このために、転送回路２０３を用いた命令スケジューリングでは実質的に命令実行可能判定や実行順序を制御する回路が不要となり、命令スケジューラ１０４よりも高速処理を実行できる。

【００２０】このように、プロセッサ２００は、スーパースカラープロセッサ１００内命令スケジューラ１０４のスケジュール結果を使用して、従って、このようなスケジュールを実行するよりも速いクロックで命令を実行する。なお、二つのプロセッサ１００、２００は必ずしも物理的に近接している必要性はないが、信号線３０１～３０５が通常のプロセッサ内部信号に匹敵する動作速度を必要とするため、例えば同一シリコン基板上に形成されるなどが好ましい。

【００２１】（２）命令の実行態様

スーパースカラープロセッサ１００内の命令フェッチ回路１０２は、命令デコーダ１０３へ送出した命令のアドレスを命令アドレス３００として翻訳命令記憶領域２１１にも供給する。図２を参照するに、プロセッサ２００では、この命令に対する翻訳命令が翻訳命令メモリ２０１に記憶されているかを検索する（ステップ２２１）。もし、その命令アドレスに一致する命令ラインが翻訳命令メモリ２０１にあると（ステップ２２２）、そのラインに記憶された翻訳命令を読み出す（ステップ２２３）。翻訳命令デコーダ２０２は、読み出された翻訳命令をデコードし、その命令に含まれた転送元物理資源の情報を転送回路２０３内部の位置情報へ変換する（ステップ２２４）。転送回路２０３は実行に必要なデータが揃った命令から順に演算器２０５，一時レジスタ２０６、レジスタＩＯ２０７へデータを転送する。データを受け取った演算器２０５は演算結果を再び転送回路２０３へ戻し，演算命令が順次実行される（ステップ２２５）。ステップ２２２で翻訳命令記憶領域２１１内に命令アドレス３００と同一の命令開始アドレスが発見されなかった場合には、プロセッサ２００は作動せず、スーパースカラープロセッサ１００による命令実行が実施される。すなわち、命令デコーダ１０３にて命令がデコードされ（２２６）、命令スケジューラ１０４で命令の実行順序が決定された（２２７）後、演算器１０５にて命令が実行される（２２８）。

【００２２】（３）スーパースカラープロセッサ１００

このプロセッサの概要は以下の通りである。命令キャッシュ１０１に格納された命令は、命令フェッチ回路１０２にてフェッチされ、命令デコーダ１０３に送り込まれる。命令デコーダ１０３は命令語が示す入出力レジスタの番号や演算種別の情報を解読する。命令フェッチ回路１０２，命令デコーダ１０３は、いずれも複数の命令（一般的なスーパースカラープロセッサの能力として４～８命令）を同時に処理できるものとする。デコーダ１０３は、いわゆるレジスタリネーミング技法を用いて、プログラムで指定可能な論理的レジスタ番号をプロセッサ内部にある多量な物理レジスタ番号に変換する機能を持っていても良い。

【００２３】命令スケジューラ１０４は命令デコーダ１０３から送られてきた複数の命令について、命令が利用する演算器の割り当てと命令実行順序の決定を行う。より具体的には、本実施の形態では、分岐命令処理ユニットも一つの演算器として扱われる。複数の演算器１０５には、整数演算器、浮動小数点演算器、分岐命令処理ユニットのように機能が異なるものおよび機能が同一の複数の演算器が混在している。演算器の個数はプロセッサで並列実行可能な命令数を制限するが、本実施の形態は、特定の数の演算器に限定はされない。いずれかの演算器１０５に対して複数の入力データラッチが存在する場合には、それぞれの入力データラッチに対して一意に識別可能な番号が付けられているものとする。この番号としては、例えばその入力データラッチに供給すべき演算結果データを出力する演算器の識別番号の下位ビット側に入力データラッチの番号を付加したものを利用できる。

【００２４】命令の実行順序は命令実行に必要な入力データが揃った命令から順に実行することが原則となる。この制御には例えばＴｏｍａｓｕｌｏアルゴリズムを利用することができる。命令スケジューラ１０４には実行完了を待つべき先行命令が存在する命令が停留し、命令実行に必要な入力データが全て揃った命令は演算器１０５へ送出される。命令が完了するとレジスタの値が確定したという意味で値の確定したレジスタ番号がレジスタファイル１０７から命令スケジューラ１０４へ伝えられる。命令スケジューラ１０４は、停留している命令の中

に、伝えられた番号のレジスタを入力値として使用する命令があるか否か、入力値が全て揃った命令があるかを調べ、そのような命令があればその命令をいずれかの演算器１０５へ送出する。

【００２５】以上説明した命令スケジューラ１０４の動作によって命令の実行順序が動的に決定され、その結果として各命令が使用した演算器、および命令の入力値となるデータを演算した演算器が決定できる。したがって、演算器やレジスタなどの物理資源を識別する番号を物理資源番号と定義すると、命令の動的な実行履歴はある物理資源からある物理資源へのデータ転送の履歴として見ることができる。命令スケジューラ１０４は、演算器に送付した命令の命令アドレスである実行命令アドレス３０１、その命令によって起こるデータ転送の転送先と転送元をそれぞれ転送先物理資源３０２、転送元物理資源３０３などの命令実行履歴として命令翻訳回路２０４に送出する。

【００２６】以下では、スーパースカラープロセッサ１００の回路の詳細をさらに説明する。命令フェッチ回路１０２は、フェッチした命令のアドレス３００を命令デコーダ１０３とプロセッサ２００に供給する。このようにプロセッサ２００が実行すべき命令のアドレスを命令フェッチ回路１０２より取り出すと、プロセッサ２００が実行すべき命令列をできるだけ早い時期に取り出すことになり、それ以降のスーパースカラープロセッサ１００がその命令に対して処理を実行するのを回避することができる。なお、実装面積等の制約により命令アドレス３００を命令フェッチ回路１０２より取り出すことができない場合には、命令デコーダ１０３や命令スケジューラ１０４、演算器１０５といった命令実行ステージにより近い場所から命令アドレス３００を取り出すことも可能である。命令フェッチ回路１０２は命令列の命令キャッシュ１０１からの命令の獲得を滞りなく実行するために、分岐命令の予測機能、命令バッファ機能等によって、実行可能性のある命令は先行的かつ余分に獲得するそれ自体公知の機能を有する。したがって、命令フェッチ回路１０２から命令デコーダ１０３へ送出された命令は情報処理装置で実行することが確定した命令列と考えることができる。

【００２７】本実施の形態では図４に示したフォーマットの命令を用いる。この命令フォーマットはいわゆるＲＩＳＣプロセッサで用いられる典型例であり、レジスタ番号等は特別な解釈なしに、命令のビットフィールドを分割するだけで得ることができる。Ｘフォーマットは一般の算術論理演算命令に用いられるフォーマットであり、冒頭の命令種を表すオペコード（ＯＰＣＤ）フィールド、演算結果が格納されるレジスタ番号であるターゲットレジスタフィールド（ＲＴ）、演算の入力数値が格納されるレジスタ番号として二つのオペランドレジスタフィールド（ＲＡ，ＲＢ）および拡張機能情報フィールド（ＥＯ）から構成される。Ｄフォーマットはロード、ストア命令のフォーマットである。オペコードフィールド、ターゲットレジスタフィールド、オペランドレジスタフィールドはＸフォーマットと同様の機能を持ち、ディスプレースメントフィールド（Ｄ）はロード、ストア命令のアクセスするメモリ番地計算用の加算値を格納する。メモリ番地はオペランドレジスタ（ＲＡ）の値とディスプレースメントフィールドの値の加算値となる。オペコード（ＯＰＣＤ）フィールドから命令実行で利用される演算器の種類が判別でき、レジスタに関するＲＴ，ＲＡ，ＲＢフィールドから、命令が利用するレジスタが決定される。

【００２８】図３において、スーパースカラープロセッサ１００内の命令デコーダ１０３は、命令フェッチ回路１０２から送られてきた命令を図４に例示した命令フォーマットにしたがって分割する。つまり、ラッチ１１０～１１３はそれぞれオペコード（ｏｐｃｄ），ターゲットレジスタ（ＲＴ）番号１１７，オペランドレジスタ（ＲＡ）番号１１８と、最後にオペランドレジスタ（ＲＢ）の番号またはディスプレースメント（Ｄ）の値を保持する。オペコードデコード回路１１５は、ラッチ１１０が保持するオペコードをデコードし、オペランドレジスタ（ＲＢ）を使用するか否かの判定信号１１４と命令で使用する演算器の物理リソース番号である演算器番号（ＦＵ）１１６とを生成する。シフト回路１２２は判定信号１１４がＨｉｇｈレベルのとき、命令フィールドの１６－２０ビットを下位２７－３１ビットにシフトし、上位１６－２０ビットは１を埋め直し、拡張データ（ＥＸ）１１９として出力する。判定信号１１４がＬｏｗのときには、命令フィールドの１６－３１ビットのディスプレースメント情報がそのまま拡張データ（ＥＸ）１１９として出力される。

【００２９】命令スケジューラ１０４は、書き込みスコアボード１２０、読み出しスコアボード１２１および命令実行バッファ０～命令実行バッファ３を用いて先に述べた概念の命令スケジューリングを実行する。書き込みスコアボード１２０と読み出しスコアボード１２１はどちらもレジスタ番号をアドレス情報とするメモリである。書き込みスコアボード１２０はメモリデータとして各レジスタ番号毎に２ビットの領域を持ち、書き込もうとしているレジスタを読み出そうとしている先行命令が命令実行バッファ０から命令実行バッファ３までに幾つ存在するかを示すカウンタとなる。読み出しスコアボード１２１はメモリデータとして各レジスタ番号毎に１ビットの領域を持ち、演算器で実行中の命令の中に読み出そうとしているレジスタ内容を更新しようとする命令があるか否かを示す。命令実行バッファ０から命令実行バッファ３は先入れ先出し（ＦＩＦＯ）キューを構成しており、命令デコーダ１０３の出力である、演算器番号１１６，ターゲットレジスタ番号１１７，オペランドレジ

スタ番号１１８，拡張データ１１９の各信号と、命令フェッチ回路１０２からの命令アドレス１２５を保持する。命令選択回路１２３は、上記のＦＩＦＯキューに格納されている命令の内、実行可能で、最もＦＩＦＯキューに入ってから時間の経つものを選択する。

【００３０】命令スケジューリングの詳細は次の通り、命令デコーダ１０３から送られてきた各レジスタ番号の情報は書き込みスコアボード１２０と読み出しスコアボード１２１とに入力される。その間に命令は命令実行バッファ０に取り込まれる。ただし，ヒット信号２４２がＨｉｇｈのときは翻訳命令が実行されるため，その間命令デコーダ１０３から送られてくる命令のスケジューリングおよび以降の動作はキャンセルされる。上記両スコアボードの出力値が０のときは、命令で使用するターゲットレジスタもオペランドレジスタも先行する命令によって使用中ではないことになる。つまり当該命令は実行可能であることがわかる。この段階で実行可能でない命令は、いずれかのレジスタの状態が変化するのを待つ必要があり、後続の命令が命令デコーダ１０３から到着するに従ってＦＩＦＯキューのより深い位置（命令実行バッファ１～３）へ進む。命令実行バッファ０～３には常に書き込みスコアボード１２０と読み出しスコアボード１２１の出力信号とレジスタファイル１０７からの演算終了信号１２４が放送回路１２５を介して通達される。この通達により、オペランドレジスタへの書き込みが終了し、かつ先行命令によるターゲットレジスタの読み出しが完了した命令が実行可能となる。

【００３１】命令選択回路１２３は命令実行バッファ０～３の中で実行可能となった命令を取り出し、オペランドレジスタの番号をレジスタファイル読み出し回路１２６へ、命令アドレス、ターゲットレジスタ番号、演算器番号、拡張データを命令発行回路１２７へ送出する。レジスタファイル読み出し回路１２６ではオペランドレジスタの内容をレジスタファイル１０７（図１）から読み出し、オペランドデータとして命令発行回路１２７へ送り出す。

【００３２】命令発行回路１２７は演算器番号で識別される演算器１０５にターゲットレジスタ番号、オペランドデータ、拡張データを送り出す。書き込みスコアボード１２０のメンテナンス動作としては、命令デコーダ１０３から新規に命令実行バッファ０に到着した命令のオペランドレジスタ（ＲＡ，ＲＢ両方）の番号が書き込みスコアボード１２０のセット端子（Ｓ）に送られ、該当レジスタのカウンタ値がインクリメントされる。また、実行可能となった命令のオペランドレジスタ（ＲＡ，ＲＢ両方）の番号はレジスタファイル読み出し回路１２６から書き込みスコアボード１２０のリセット（Ｒ）端子に送られ、該当レジスタのカウンタ値が０になる。読み出しスコアボード１２１はレジスタ更新をする命令が発行されたときに命令発行回路１２７からターゲットレジスタ番号の通知をセット端子（Ｓ）に受け、更新中を示すフラグ１を立てる。このフラグは演算終了信号１２４によってリセットされる。先に述べた翻訳命令の概念から実行可能となった命令の命令アドレスが実行命令アドレス３０１として、演算器番号とターゲットレジスタ番号とが転送先物理資源番号３０２として、オペランドレジスタ番号と拡張データが転送元物理資源番号３０３として命令翻訳回路２０４へ送られる。以上の命令スケジューリング回路の動作は，仮に命令デコーダ１０３の動作にパイプライン１ステージ分の時間がかかるとすると，少なくとも２ステージ以上の時間を要する処理となる。

【００３３】（４）プロセッサ２００

（４ａ）プログラム例

以下の説明では図５に示したプログラム例を用いる。図５のプログラム例は５種の命令から構成されており、ｌｆｄ命令５０１は浮動小数点数値を６番レジスタに格納されている数値とディスプレースメントである８で示されるメモリ番地からロードし、浮動小数点レジスタ１に格納する。ａｉ命令５０２は６番レジスタの値に４を加算し、再び６番レジスタの値とする。ｆｍ命令５０３は浮動小数点レジスタ１番と２番に格納された数値の積を再び浮動小数点レジスタ１番に格納し、ｆｃｍｐ命令５０４は浮動小数点レジスタ１番の数値が０番の数値よりも大きいか小さいかを比較し、その比較結果を条件レジスタ６番（ＣＲ６）に格納する。最後のｂｃ命令５０５は分岐命令であり、ｆｃｍｐ命令５０４の結果、浮動小数点レジスタ１番の数値が０番のレジスタの数値よりも小さいとき、ラベル＿Ｌ１０に分岐する。

【００３４】この命令列の参照、更新するレジスタと命令の関係をグラフ理論的に図示したものが図６である。この図は命令実行過程を概念的に表すものであり、本実施の形態の装置内の機構と直接対応するものではない。図６からわかるように、グラフ節点６０１，６０２，６０３で示すレジスタは先行命令で更新された後、ただちに後続の命令で参照されており、もし先行命令が更新するレジスタと値を必要としている演算器の両方へ同時に実行結果を転送可能であると、全体の実行時間を短縮できる。このような実行制御方法はフォワーディングと呼ばれ、命令スケジューラ１０４が実行前の命令列を解読することによって、転送路を決定する。つまり、スーパースカラープロセッサ１００で図５の命令列を実行すると、実行履歴は図７に示したグラフに対応することになる。

【００３５】上記のフォワーディングにより、演算結果は最短経路で転送するようにスケジューリングされており、図７の上方および下方には図５の命令列に対する入力値を格納したレジスタ番号（レジスタ名）と出力結果が格納されたレジスタ番号（レジスタ名）が並ぶ。このグラフで命令種を示す節点を演算器と読みかえると、グ

ラフの枝はレジスタと演算器または演算器と演算器のデータ転送を表していることになる。

【００３６】（４ｂ）分岐ブロックの生成

本実施の形態での翻訳命令は、この枝が表すデータ転送の集合であり、枝の横に枝を識別するために付した丸数字が翻訳命令の最小単位である。また、図７において波線で囲んだブロックが示すように図５に示した一群の命令列は、入力値となるブロック６７０、演算内容となるブロック６７１、演算結果となるブロック６７２に分類することが可能である。このことはブロック６７１を一つのマクロ命令として考えたときに、ブロック６７０がその実行に対して初期化条件に、ブロック７６２が終了条件に対応していると考えられる。以上の概念に基づき、命令翻訳回路２０４では次の手順に従って翻訳命令を作成する。

【００３７】入力信号は先に説明した実行命令アドレス３０１，転送先物理資源番号３０２，転送元物理資源番号３０３であり、出力は図１１に示した翻訳命令３種類、すなわち初期化命令７０１，翻訳命令本体７０２，終了命令７０３である。また、翻訳命令は一組の初期化命令７０１と終了命令７０３と、それらに挟まれた１つまたは複数個の翻訳命令本体７０２で１単位を成す。最初に、翻訳命令生成回路２０４は１命令実行毎に送られてくる入力信号を次の基準で分岐ブロックに分割する。この分岐ブロックに含まれる複数の命令が翻訳命令の１単位に対応する。

【００３８】翻訳命令生成回路２０４は転送先物理資源番号３０２を観測し、その転送先が分岐命令処理ユニットであるとき分岐命令が実行されることを認識する。分岐ブロックは分岐命令と分岐命令に挟まれる命令列であり、図５に示した例のように、非分岐命令から始まり、分岐命令で終了する。ただし、ここでは命令実行結果に基づく動的な命令実行順序であるので、例えばアセンブラ言語として図５の命令５０１の一つ上が分岐命令であるか否かは無関係である。つまり、プログラム中ある分岐命令によって実行アドレスがかわり、命令５０１にジャンプした場合、命令５０１から命令５０５までが一つの分岐ブロックを形成する。また、仮に分岐命令５０５の条件が成立せず命令５０１に分岐しなくても、分岐命令５０５の次に実行される命令から新しい分岐ブロックが開始する。したがって、この定義からプログラム中のある命令が複数の分岐ブロックに属する場合もあり得る。さらにプログラム全体は複数の分岐ブロックにて形成されているので，実行されたプログラム全体が順次翻訳命令に変換されると考えられる。

【００３９】（４ｃ）翻訳命令

翻訳命令生成の機能は二つの機能に大別される。一つは翻訳命令本体７０２の生成である。図１１に示すように、翻訳命令本体７０２のフィールドはデータ転送の転送先と転送元から構成されており、スーパースカラープロッセサ１００から出力される転送先物理資源番号３０２を，転送先フィールド７０４に転送元物理資源番号３０３を格納すればよい。翻訳命令本体７０２の命令長は有限であり、その命令長を越える場合には、新しい翻訳命令本体７０２を別途生成する。大別した機能の２番目は初期化命令７０１および終了命令７０３の生成である。命令翻訳回路２０４には、後に説明するように、図１２に示したスコアボード７５１と意味的に等価なものが備えられている。スコアボードはスーパースカラープロセッサ１００に内蔵されている個々のレジスタについて、そのレジスタに書き込み動作を実施した演算器の番号と読み出し動作を実施した演算器番号が記録される。なお、分岐ブロックが変わった際には、スコアボード全体が０クリアされる。この番号の記録はスーパースカラープロセッサ１００の命令実行順序に従うため、分岐ブロックが終了した時点では最近に読み出し、あるいは書き込みを実施した演算器の番号が記録されている。

【００４０】本実施の形態では翻訳命令を実行するプロセッサがスーパースカラープロセッサ１００とは独立して存在し、レジスタとしては一時レジスタ２０６を用いる。そのために、一時レジスタ２０６の初期化が必要である。翻訳命令生成回路２０４はスコアボード７５１の中で読み出しはされるが、書き込みが行われていないレジスタを抽出し、初期化命令７０１を生成する。つまり、上記の条件を満たすレジスタは図７のブロック６７０のように、着目分岐ブロック外で値が設定されているレジスタである。初期化命令７０１の転送先フィールドはプロセッサ２００の内部に設けられた一時レジスタの番号であり、転送元はレジスタＩＯ２０７の物理資源番号である。レジスタＩＯ２０７はスーパースカラープロセッサ１００のレジスタファイル１０７と一時レジスタ２０６間のデータ転送機能を有する。なお、初期化命令７０１の命令開始アドレスには分岐ブロックの先頭命令のアドレスが格納される。

【００４１】終了命令７０３も同様にスコアボード７５１を参照して生成される。終了命令７０３は分岐ブロックが終了した際の結果を一時レジスタ２０６からレジスタファイル１０７に格納するために存在する。終了命令７０３の転送先フィールドはレジスタＩＯ２０７の物理資源番号であり、転送先フィールドは分岐ブロックが終了した時点でスコアボード７５１に書き込み実績が残っているレジスタ番号である。以上のようにして生成された翻訳命令は分岐ブロック終了と同時に、翻訳命令メモリ２０１内の書き込み回路２１４へ転送される。以上が翻訳命令の生成の概略説明である。

【００４２】（４ｄ）命令翻訳回路２０４

図８に示すように、命令翻訳回路２０４は、スコアボード７５１ａ，７５１ｂと初期化命令バッファ１３０，翻訳命令本体バッファ１３１，終了命令バッファ１３２等から構成される。スコアボード７５１ａと７５１ｂは、

それぞれ図１２に示したスコアボードのＲｅａｄ部とＷｒｉｔｅ部に記載された情報を保持するスコアボードであり、それぞれこれらのＲｅａｄ部とＷｒｉｔｅ部に保持された演算器番号を保持する。分岐検出回路１３３は演算器番号が分岐命令処理ユニットであるとき、リセット信号１３４と翻訳命令送出要求信号１３３ａを生成する。これにより、先に述べた分岐ブロックが認識できる。リセット信号１３４を受けたスコアボード７５１ａ，７５１ｂは、それぞれの記憶内容を全て０クリアする。

【００４３】翻訳命令本体の生成は命令スケジューラ１０４から実行命令アドレス３０１、転送先物理資源番号３０２、転送元物理資源番号３０３が送られる毎に実施される。つまり、図１２を用いて説明した概念のとおり、転送元の物理リソースは、スコアボード７５１ｂに演算器番号が格納されていれば、その演算器であるし、演算器番号がスコアボード７５１ｂに格納されていなければオペランドレジスタが転送元となる。その選択は選択回路１３５が実施する。また、転送先はターゲットレジスタと演算器のいずれかになる。したがって、２オペランド１ターゲットの命令からは最大３組の転送元と転送先が生成され、それらは翻訳命令バッファ１３１に格納する。初期化命令と終了命令はリセット信号１３４に同期して生成される。リセット信号１３４は０検出回路１３６ａと１３６ｂに入力され、それぞれのスコアボード内の格納値が０であるレジスタ番号を出力する。スコアボード７５１ａの０検出回路１３６ａの出力は、終了命令の転送先を指定し、終了命令バッファ１３２へ格納される。 スコアボード７５１ｂの０検出回路１３６ｂの出力は、条件判定回路１３７においてスコアボード７５１ａの出力結果と合成される。条件判定回路１３７の動作内容は、０検出回路１３６ｂから出力された書き込み実績の無いレジスタ番号のうち、読み出し実績のあるレジスタ番号を選別することである。条件判定回路１３７から出力されるレジスタ番号は初期化，すなわちレジスタファイル１０７の内容を一時レジスタ２０６へ転送する必要の番号である。プロセッサ２００の初期化命令はレジスタＩＯ２０７を通してレジスタファイル１０７の内容が転送されるので，転送元がレジスタＩＯ２０７，転送先が一時レジスタ２０６となる。この初期化命令は初期化命令バッファ１３０へ格納される。最後に翻訳命令送出回路１３８は初期化命令、翻訳命令本体、終了命令をこの順で結合し、翻訳命令メモリ２０１へ出力する。

【００４４】（４ｅ）翻訳命令メモリ２０１

図９を参照するに、翻訳命令メモリ２０１は、書き込み回路２１４と、翻訳命令記憶領域２１１と、読み出し回路２１３とからなる。なお、図には翻訳命令デコーダ２０２の内部構造も併せて示す。翻訳命令記憶領域２１１は、ＭＯＳトランジスタからなる複数のメモリセル２３

３および２３４と、付随する制御回路（Ａ）２３５と、制御回路（Ｒ）２３６およびその他の回路とから構成されるメモリアレイである。翻訳命令記憶領域２１１はより多くのメモリセルが繰り返し存在し、全体としてのセルアレイを構成しているが、それらのメモリセルは簡単化のために図示していない。翻訳命令記憶領域２１１は、翻訳命令に含まれる命令開始アドレスを格納する複数のメモリセル（Ｍ１）２３３と、転送元となる物理資源番号を格納するための複数のメモリセル（Ｍ）２３４とから構成されている。それぞれのメモリセルは１ビットのセルで、それらの情報を保持するには翻訳命令記憶領域２１１に設けられた複数ビット分のメモリセルが使用されるが、以下では、簡単化のため必要なビット数分全てのメモリセル２３３，２３４は図示しない。また便宜上、水平方向に並ぶメモリセル群をライン、垂直方向に並ぶメモリセル群をカラムと呼ぶ。

【００４５】書き込み回路２１４は、命令翻訳回路２０４から線２１９を介して供給される翻訳命令がまだこの翻訳命令メモリ２０１に記憶されていないときには、この翻訳命令を翻訳命令記憶領域２１１に格納する。その際、その翻訳命令が、図１１に示した初期化命令７０１であるときには、その命令７０１は、翻訳命令記憶領域２１１内のいずれか一つのラインに記憶される。この命令内の各転送元は、転送先となり得る複数の物理資源に一対一に対応し、その対応する物理資源に対して定められた水平方向の記憶位置に記憶される。すなわち、その翻訳命令７０１中の複数の転送先の各々に対応する水平方向の位置に、その転送先と対をなす転送元を記憶する。このために、書き込み回路２１４では、カラムデコーダ２７０が翻訳命令中の各転送先をデコードし、翻訳命令記憶領域２１１のｘ方向位置を算出する。書き込み増幅器２７１が、算出された位置にその転送先と対をなす転送元データを書き込む。命令翻訳回路２０４から順次与えられる複数の翻訳命令は、翻訳命令記憶領域２１１内のまだ命令が記憶されていないカラムに順次書き込まれる。このような書き込みは、書き込み制御回路２１２の制御下で行われる。翻訳命令によっては同じ転送先を複数含む場合がある。このような翻訳命令を翻訳命令記憶領域２１１に格納するときには、転送元情報を複数のラインに分けて翻訳命令記憶領域２１１に記憶する必要がある。その結果、それらの翻訳命令では、図１３と異なって、転送元が稠密に詰まってはいない状態として記憶される。

【００４６】命令翻訳回路２０４から与えられた命令が初期化命令７０２のときには、転送先がデコードされ，翻訳命令７０２Ａとして翻訳命令記憶領域２１１へ格納される。命令翻訳回路２０４から与えられたこの翻訳命令が終了命令７０３であるときには、その命令が指定する最後の転送元を翻訳命令記憶領域２１１へ書き込んだラインのメモリセル（Ｍ１）２３３には全て１が書き込

まれる。なお、翻訳回路２０４から与えられた命令がこの翻訳メモリ２０１に記録されているか否かは、後に説明するように、ヒット信号２４２により書き込み制御回路２１２に伝えられる。もし、この翻訳命令がヒットしなかったときにはこのヒット信号２４２がＬｏｗになる。

【００４７】各メモリセル（Ｍ１）２３３は、内容の一致検索機能を持ち、命令フェッチ回路１０２が生成する命令アドレス３００とそのメモリセル内に保持しているデータの一致照合を行う。ある、ラインの命令開始アドレス、つまりメモリセル（Ｍ１）２３３の内容が全て一致した場合には、信号線２３０の電位がＬｏｗとなり、制御回路（Ａ）２３５が読み出しライン選択信号２１６をＨｉｇｈに引き上げる。読み出しライン選択信号２１６がＨｉｇｈになると該当するラインのメモリセルは、メモリセル（Ｍ１）２３３も，メモリセル（Ｍ）２３４も保持していた論理値をデータ出力線２１８へ出力する。この読み出し動作に関してカラム方向の選択は行わないため、１ライン分のデータが読み出し回路２１３に到達する。つまり、翻訳メモリ２０１は命令開始アドレスをタグ情報とした一種のＣＡＭ（Ｃｏｎｔｅｎｔｓ Ａｓｓｏｃｉａｔｉｖｅ Ｍｅｍｏｒｙ）として機能する。

【００４８】最も基本的なメモリセル（Ｍ）２３４は、図１０（ｂ）に示すように、トランジスタＴ１とトランジスタＴ２のゲート間容量に電荷を蓄積する。書き込みライン選択信号２１５がＨｉｇｈのとき、トランジスタＴ１が開き、データ入力線２１７の値がこのゲート間容量に記憶される。読み出しライン選択信号２１６がＨｉｇｈになると、トランジスタＴ３が開き、記憶していた情報が負論理でデータ出力線２１８に現れる。データ出力線２１８は、カラム毎に設けられたプリチャージ用トランジスタ２３８（図９）によって、メモリセルの読み出し前にＨｉｇｈレベルへ引き上げられる。

【００４９】メモリセル（Ｍ１）２３３は、図１０（ａ）に示すように、トランジスタＴ４，Ｔ５，Ｔ６はそれぞれメモリセル（Ｍ）２３４のトランジスタＴ１，Ｔ２，Ｔ３に対応し、同等な役割を持つ。セル内の２つのインバータはそれぞれデータ入力線２１７と命令アドレス３００の反転信号を生成するために存在するが、これらをセル内に記載したのは図面の記載の簡単化のためであり、本来は、カラム毎にこれらの一組のインバータが存在すればよい。トランジスタＴ７とＴ４およびトランジスタＴ８とＴ９の結線にはそれぞれ書き込んだデータの正論理値と負論理値の両方が記憶され、トランジスタＴ１０のゲートには、このセルに記憶されているデータと命令アドレス３００の排他的ＮＯＲの論理値が最終的に生成される。つまり、照会情報として与えられた命令開始アドレスがこのセルに記憶されている情報と一致した場合、信号線２３０はこのメモリセル（Ｍ１）２３３内で導通状態となる。

【００５０】制御回路（Ａ）２３５は、図１０（ｄ）に示す回路からなる。クロックの立ち上がりより前は、トランジスタＴ１４が信号線２３０を電源電位にプリチャージする。先に述べたように、メモリセル（Ｍ１）２３３内のトランジスタＴ１０が全て導通しているときは、クロックの立ち上がり以降、信号線２３０はＬｏｗとなる。読み出し回路２１３が与える強制的読み出し要求（Ｉｎｓｔｒｕｃｔｉｏｎ ｒｅｑｕｅｓｔ）２３７は、後述する動作の時以外にはＬｏｗレベルである。このため、メモリセル（Ｍ１）２３３で検出された一致は、ゲートｇ１を経て読み出しライン選択信号２１６をＨｉｇｈに設定することになる。このようにして、翻訳命令記憶領域２１１からは、命令アドレス３００に一致したラインのデータがデータ出力線２１８に現れる。

【００５１】制御回路（Ｒ）２３６は、リフレッシュ制御２４０，ラインデコーダ２３９と協調してメモリセルのリフレッシュを実施する回路で、図１０（ｃ）に示す構造を有する。リフレッシュ時にはリフレッシュ制御２４０がリフレッシュ信号２３２をＨｉｇｈレベルにした状態で通常の読み出し動作を実行する。つまり、ラインデコーダ２３９がリフレッシュするラインの読み出しライン選択信号２１６をＨｉｇｈにする。制御回路（Ｒ）２３６のトランジスタＴ１１とＴ１２のゲート間にデータが保持される。次にラインデコーダ２３９は書き込みライン選択信号２１５をＨｉｇｈレベルに上げ、リフレッシュ制御２４０がリフレッシュ信号２３２をＬｏｗにすることで、制御回路（Ｒ）２３６に保持されていたデータはプリチャージインバータを経て元のラインに書き戻される。また、ＯＲゲート２４１は全読み出しライン選択信号の論理的ＯＲをとることにより、先に述べた命令開始アドレスの検索におけるヒット信号２４２を生成する。

【００５２】こうして、命令翻訳回路２０４から初めて翻訳メモリ２０１に供給された命令に対してもこのヒットチェックが行われ、その命令がヒットしなかったときには先に記載したように、その命令がこの翻訳メモリ２０１に書き込まれる。さらに、すでにその命令が翻訳メモリ２０１に書き込まれた後に、再度実行されたときには、その命令が属する命令ブロックの先頭の命令の命令アドレスに対するヒットチェックの結果、その先頭の命令がヒットする。その結果、この命令から始まる一連の命令に対する複数の翻訳命令が順次翻訳メモリ２０１から読み出されることになる。

【００５３】読み出し回路２１３には、転送先となり得る複数の物理資源に対応して、複数のＦＩＦＯキュー２６０が設けられ、各ＦＩＦＯキュー２６０は、センスアンプを介して対応する物理資源に対して設けられたデータ出力線２１８に接続されている。翻訳命令記憶領域２１１から転送先データ出力線２１８に読み出されたアナ

ログ信号は、このセンスアンプで論理信号に確定された後、ＦＩＦＯキュー２６０へ書き込まれる。この書き込みは、ラッチ２６１にオール１が読み出されるまで、すなわち、終了命令が読み出されるまで繰り返される。このとき、ラインアドレスのインクリメントはＦＩＦＯキュー２６０からラインデコーダ２３９へ伝えられる。

【００５４】（４ｆ）翻訳命令の実行態様

図９において、翻訳命令デコーダ２０２は、翻訳命令の各転送元フィールドをデコードする複数の部分デコーダ２０２Ａと命令待機ユニット８０２からの転送要求であるｉｎｓｔｒｕｃｔｉｏｎ ｒｅｑｕｅｓｔ ８０８から命令読み出し回路２１３内部のＦＩＦＯキュー２６０への転送要求信号２０２ｃを生成するORゲート２０２Ｂとからなり、各部分デコーダ２０２Ａに対して、全ての物理資源、すなわち、複数の演算器２０５、複数の一時レジスタ２０６、レジスタＩＯ２０７の全てに対応する複数のｉｎｓｔｒｕｃｔｉｏｎ信号８０１が設けられている。先に述べたように、翻訳命令メモリ２０１内部では命令が要求する転送先物理資源の番号は、翻訳命令記憶領域２１１内の位置情報として記憶されているので、翻訳命令デコーダ２０２は、転送元物理資源のみをデコードすればよい。つまり、各部分デコーダは、読み出された翻訳命令の中の対応する転送元フィールドをデコードし、その転送元物理資源に対応する一つのｉｎｓｔｒｕｃｔｉｏｎ信号を選択して起動する。このデコード結果は、その部分デコーダが対応する転送先となる物理資源、すなわち、演算器２０５、一時レジスタ２０６またはレジスタＩＯ２０７のいずれか一つが待つべきデータがどの転送元の物理資源で生成されるのかを示している。各ｉｎｓｔｒｕｃｔｉｏｎ信号８０１は１ビットのデータ線から構成される。

【００５５】図１４において、転送回路２０３は、全ての物理資源、すなわち、複数の演算器２０５、複数の一時レジスタ２０６およびレジスタＩＯ２０７の各々の各入力端に対応して、その物理資源に対応する部分デコーダの複数のｉｎｓｔｒｕｃｔｉｏｎ信号８０１と、それぞれのｉｎｓｔｒｕｃｔｉｏｎ信号８０１が対応する物理資源の出力線との交点に命令待機ユニット８０２が設けられている。各物理資源の各入力端に対応して設けられた複数の命令待機ユニット８０２の出力は、ワイアドオアされて、その入力端に供給される。

【００５６】命令待機ユニット８０２の内部構成を図１５に示す。ｉｎｓｔｒｕｃｔｉｏｎ信号８０１は先に述べた翻訳命令デコーダ２０２で生成された信号であり、着目している命令待機ユニット８０２の関係する転送元と転送先との間にデータ転送が必要であることを意味する信号である。

【００５７】ｉｎｓｔｒｕｃｔｉｏｎ ｒｅｑｕｅｓｔ ８０８は命令待機ユニット８０２から翻訳命令デコーダ２０２へのデータ転送要求であり、翻訳命令デコーダ２０２内のゲートでOR論理がとられ、読み出し回路２１３内のＦＩＦＯキューから新たな転送元データが取り出される。ｄａｔａ－ｉｎ信号８０３は所定のデータ幅、例えば３２ビット幅のデータ線であり、転送元からデータを伝送する。ｄａｔａ－ｉｎ信号８０３に新しいデータが到着したことはｄａｔａ－ｉｎ－ｖａｌｉｄ信号８０５が有効になることで示される。同様に、ｄａｔａ－ｏｕｔ信号８０４は転送先へのデータ転送線であり、ｄａｔａ－ｉｎ信号８０３と同様に３２ビット等の幅が用意される。ｄａｔａ－ｏｕｔ信号８０４の有効性はｄａｔａ－ｏｕｔ－ｖａｌｉｄ信号８０６によって示される。命令待機ユニット８０２の基本的な役割はｉｎｓｔｒｕｃｔｉｏｎ信号８０１が有効、すなわちデータ転送が必要な命令が存在するときに、ｄａｔａ－ｉｎ信号８０３の内容をｄａｔａ－ｏｕｔ信号８０４に出力することである。転送先の演算器やレジスタはそれぞれの機能を実現するために必要なデータが揃った時点で動作を開始する。

【００５８】より具体的には上記の制御が転送制御回路８０７で実現される。転送制御回路８０７の回路と機能は図１６に示した回路図およびタイムチャートの通りである。つまり，ラッチ８０００とゲート８００２の組およびラッチ８００１とゲート８００３の組はそれぞれｉｎｓｔｒｕｃｔｉｏｎ信号８０１とｄａｔａ－ｉｎ－ｖａｌｉｄ信号８０５がＨｉｇｈになった状態をｄａｔａ－ｏｕｔ－ｖａｌｉｄ信号８０６がＨｉｇｈになるまで保持する。ゲート８００４は上記２つの信号がともにＨｉｇｈになった状態を検出し，ラッチ８００５の状態を１サイクルの間反転させる。このラッチ８００５の出力はｄａｔａ－ｏｕｔ－ｖａｌｉｄ信号８０６，ｉｎｓｔｒｕｃｔｉｏｎ ｒｅｑｕｅｓｔ ８０８として出力される。ｉｎｓｔｒｕｃｔｉｏｎ信号８０１が有効である場合，この命令待機ユニット８０２は転送元からのデータを待っている状態であり、次にｄａｔａ－ｉｎ信号８０３にデータが到着すると、転送先の演算器やレジスタの格納動作を起動する意味でｄａｔａ－ｏｕｔ信号８０４から同一データを出力する。図１４に戻り、ｄａｔａ－ｉｎ信号８０３およびｄａｔａ－ｉｎ－ｖａｌｉｄ信号８０５は同一水平位置にある命令待機ユニット８０２に全て接続され、電気信号は一斉に伝達するよう接続されている。したがって、先に述べたように命令が転送元と転送先に接続される結線の交差する位置で実行を待機する動作が実現できる。

【００５９】次に翻訳命令の実行形態を視覚的に把握する目的で、図１７から図２０に図５で例示したプログラムの実行例を示す。図１７から図２０において丸数字は図７の丸数字に対応しており、翻訳命令のデータ転送をあらわしている。また、丸数字の０は翻訳命令の初期化命令に、丸数字の１２は終了命令に対応している。図１７の上方に囲んだ領域９００は翻訳命令メモリ２０１に

格納された翻訳命令を表し、下方の領域９０１は図１４において命令待機ユニット８０２が設けられていた領域に対応する。領域９０１の下に並べられた図１における演算器２０５～レジスタＩＯ２０７に対応しており、説明の便宜上実行するプログラムと同じ演算名とレジスタ名で対応関係を示している。なおレジスタＩＯ２０７は図１７から図２０においては１つだけ例示しているが、これは図面の都合上であり、データ転送が必要なレジスタの数だけレジスタＩＯも存在する。また図１７中の丸数字１２下方にある黒丸は、翻訳命令の初期化命令を起動するために存在し、初期化命令毎、かつレジスタＩＯ２０７対応に存在する。また，本発明の原理から一時レジスタ２０６の物理資源番号はレジスタファイル１０７内のレジスタ番号と一致している必要がある。初期化命令によって一時レジスタにレジスタデータの初期値を転送する際，一時レジスタ２０６の物理資源番号も同時に付与され，以降の命令実行ではその物理資源番号が使用される。

【００６０】翻訳命令はｉｎｓｔｒｕｃｔｉｏｎ信号８０１が空の場合にはただちに領域９００から領域９０１の命令待機ユニット８０２へ転送される。すなわち、翻訳命令の実行は図１７の状態からただちに図１８の状態へ移行する。図１７に存在した黒丸のデータ転送命令はレジスタＩＯ２０７を起動し、レジスタファイル１０７から所望の（図７からわかるように、本プログラム例ではｆｐ０，ｒ６，ｆｐ２の初期内容）データを読み出し、レジスタＩＯの出力結果とする。図１５の命令待機レジスタＩＯ の出力線９０２上には丸数字０の命令が待機しており（図１８）、レジスタＩＯを介してレジスタファイル１０７の内容が信号線８０３に出力されると、丸数字０の直下に位置づけられたそれぞれの演算器へデータを転送し、演算を起動する。ここで信号線８０３は命令待機ユニット８０２のｄａｔａ－ｉｎ信号８０３と同一であり、各演算器、一時レジスタの出力内容は命令待機ユニット８０２へ一斉に送られる。

【００６１】以降の処理はいわゆる玉突き状態で進行する。例えば、丸数字０によってｒ６（レジスタ６番）の内容が読み出されると図１８に示すように、その出力線上に待機している丸数字１と丸数字１０に対応してａｉ（加算）とｌｆｄ（浮動小数点データのロード）を実施する演算器へｒ６のデータが転送される。ｆｐ２（浮動小数点レジスタ２番）、ｆｐ０（浮動小数点レジスタ０番）に対応した丸数字０のデータ転送命令も同様であり、結果として図１９の状態を得る。図１９の状態からは丸数字の１１によって１２が実行され、加算後のｒ６の内容がレジスタファイルに書き戻され、図２０の状態となる。最終的には丸数字１２で示したデータ転送が実施され、図７のブロック６７２が示すように、ｒ６と条件レジスタＣＲ６，浮動小数点レジスタｆｐ１の値がレジスタファイル１０７に書き込まれる。

【００６２】以上の翻訳命令実行では命令待機ユニットで待機中の命令は真に必要なデータのみを待つために、命令スケジューラ１０４で実施されていた従来技術のように演算を終了した全データと待機中の全命令との照合等が不要となり、高速な命令待機と実行が実現できる。また、この過程が従来技術より高速であるということは、同じ実行内容の命令列でも、翻訳命令メモリ２０１に蓄積された命令列の方が、従来の命令デコーダ１０３，命令スケジューラ１０４を経由する実行より高速であることを意味する。

【００６３】以上に示した実施の形態によれば、命令実行に必要な物理資源の割り当て、決定に関しては最初の命令実行時には従来技術と同様な手段により認識するため、処理時間の改善はない。しかし、２回目以降の同一命令の実行に関しては翻訳命令メモリ２０１内に格納されているデータ転送先、データ転送元の情報を利用するため、スーパースカラープロセッサ１００内の命令デコーダ１０３による処理と命令スケジューラ１０４による処理を省略することができ、その分の高速化が可能となる。実質的効果は、同じ命令列を利用する割合、すなわち翻訳メモリのヒット率に依存する。命令キャッシュなどの研究から知られているように、実行命令の局所性は高く、特に翻訳メモリ２０１を同一チップ内のＤＲＡＭ等で実現すると、ヒット率はほぼ１００％を実現することが可能である。

【００６４】また、本実施の形態では、実行すべき命令が実行に必要なデータをデータが転送されてくる経路上で待つため、リザベーションステーションのように、明らかに関連性のない命令に対して命令実行可能性のチェックを行うようなことはない。事実、従来技術では上記の冗長な機能のために、実行待ちの命令は自分の必要とするデータを番号で管理しており、演算終了時に告知される番号との照合が必要である。このことから、一つのリザベーションステーションに待機可能な命令は４命令程度が限度であり、比較にも時間を必要とした。本実施の形態では、転送先、転送元の物理資源情報は転送回路２０３中の位置情報として利用され、最終的に命令として転送回路２０３内で待機する命令は実行を起動するための１ビットの情報でしかない。このことから、実行条件の成立の判定は高速であり、構造が簡単になることから多くの命令を実行待機させることができる。また、データの転送先に対応した場所で命令実行を待機するため、存在する演算器数以上の命令を実行しようとすることもないし、それらの調停回路も必要でなくなる。

【００６５】＜変形例＞本発明は、以上の実施例に限定されるものではなく、以下に示す変形例およびその他のいろいろの変形例として、実施可能である。

【００６６】（１）実施の形態１ではスーパースカラープロセッサ１００の複数の演算器１０５、レジスタファイル１０７とは別にプロセッサ２００用に複数の演算器

２０５、複数の一時レジスタ２０６が設けられた。しかし、この後者の演算器とレジスタに代えて、前者に含まれた演算器およびレジスタを使用することも可能である。このためには、実施の形態１の転送回路２０３を、スーパースカラープロセッサ１００内の演算器１０５とレジスタファイル１０７内のレジスタの間でデータ転送するように、これらの演算器とレジスタに接続する。このときには、プロセッサ２００内の翻訳メモリ２０１、翻訳命令デコーダ２０２、転送回路２０３等の回路を動作させるクロックは、スーパースカラープロセッサ１００を動作させるクロックと同じとすることが現実的である。したがって、実施の形態１で述べたような高速な動作は実現できないが、翻訳命令を使用する結果、命令スケジューラ１０４使用しないで翻訳命令を実行できるので、この点で従来より高速な動作が期待できる。

【００６７】（２）上記の実施形態では図１１に示した翻訳命令の転送先を命令翻訳回路２０４でデコードし、翻訳命令メモリ２０１内部には図１３のように転送元だけの情報を記録していた。これは、図１４に示すように、関連するレジスタや演算器と翻訳命令メモリ内のデータを対応づけておいた方が、翻訳命令の取り出しから実行までの時間が短縮されるからである。しかし、この方法は翻訳命令メモリ２０１の利用効率が必ずしも高くないため、図１１に示した翻訳命令の形態をそのまま翻訳命令メモリ２０１内に格納することも可能である。ただし、この方法では翻訳命令読み出し時に転送先に関わるデコードをするため、本発明の主眼である、命令実行までの処理時間は若干犠牲となる。

【００６８】（３）翻訳命令メモリとしてはＳＲＡＭやＤＲＡＭといった半導体メモリを想定しているが、たとえば図２１に示すように、ハードディスクのような外部記憶装置２０１Ａに格納する形態も可能である。翻訳命令フェッチ回路２０１Ｂは読み出し回路２１３と同様に翻訳命令を蓄積する手段（本変形例では外部記憶装置２０１Ａ）から翻訳命令の転送元情報を読み出す。レジスタファイル１０７Ａはレジスタファイル１０７と論理，物理両面で同一のレジスタファイルである。ただし、この場合には上述の実施形態とは異なり、翻訳命令はスカラープロセッサ１００の動作と同時には実行されず、別途プロセッサ２００上で実行される。

【００６９】

【発明の効果】本発明によれば、逐次実行用のプログラムの命令に対してスケジュール処理を施してそれらの命令を並列に実行した後、スケジュール処理の結果をメモリに格納するので、再度そのプログラムを実行するときには、そのスケジュール処理を再度実行する必要がなく、そのプログラムをより高速に再実行できる。

【図面の簡単な説明】

【図１】本発明に係る情報処理装置の概略ブロック図。

【図２】図１の装置における命令実行のフローチャート。

【図３】図１の装置に使用されるスカラープロセッサ１００の概略構成図。

【図４】図１の装置に用いる命令のフォーマットを示す図。

【図５】図１の装置で実行されるプログラムの例を示す図。

【図６】図５のプログラムのグラフ表現を示す図。

【図７】図５のプログラムのデータ転送を表すグラフ表現を示す図。

【図８】図１の装置に使用される命令翻訳回路の概略ブロック図。

【図９】図１の装置に使用する翻訳メモリの概略ブロック図。

【図１０】図９の装置に使用される複数種のメモリセルと複数の制御回路の回路図。

【図１１】図８の装置により生成される翻訳命令のフォーマットを示す図。

【図１２】図８の装置内に含まれた二つのスコアボードに保持された内容と同じ内容を保持する等価なスコアボードを示す図。

【図１３】図９の装置に記憶された翻訳命令のフォーマットを示す図。

【図１４】図１の装置に含まれた翻訳命令デコーダと転送回路の概略構成図。

【図１５】図１４の装置に使用される命令待機ユニットのブロック図。

【図１６】図１５の装置の回路図とタイムチャート。

【図１７】図１の装置における翻訳命令の第１の実行例を示す図。

【図１８】図１の装置における翻訳命令の第２の実行例を示す図。

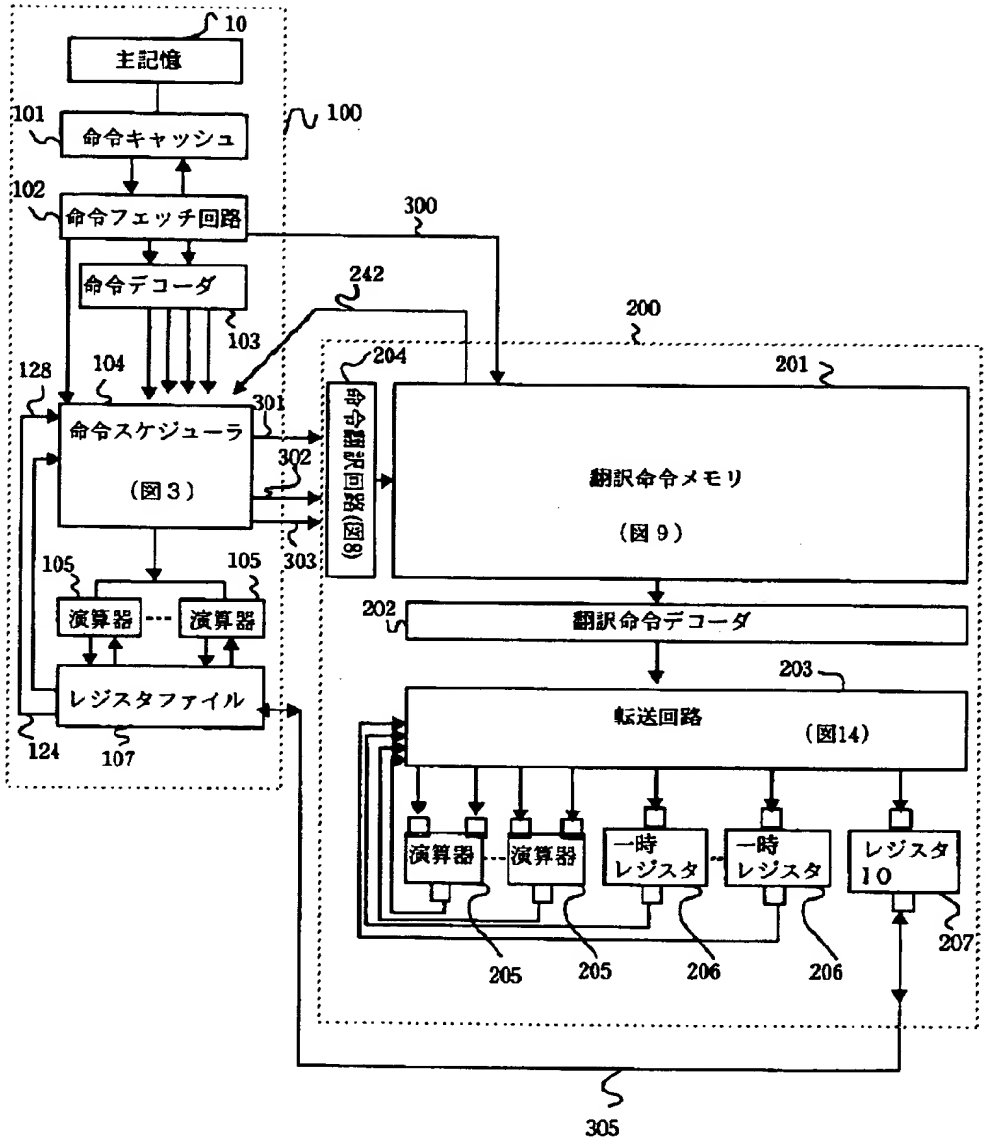
【図１９】図１の装置における翻訳命令の第３の実行例を示す図。

【図２０】図１の装置における翻訳命令の第４の実行例を示す図。

【図２１】本発明に係る他の情報処理装置の概略ブロック図。

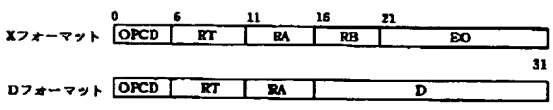
【図１】

図１


10
主記憶
100
101
命令キャッシュ
102
命令フェッチ回路
300
命令デコーダ
103
242
200
204
201
128
104
301
命令スケジューラ
（図３）
302
303
命令翻訳回路(図8)
翻訳命令メモリ
（図９）
105
105
演算器
演算器
202
翻訳命令デコーダ
203
レジスタファイル
転送回路
（図14）
124
107
演算器
演算器
一時
レジスタ
一時
レジスタ
レジスタ
10
207
205
205
206
206
305

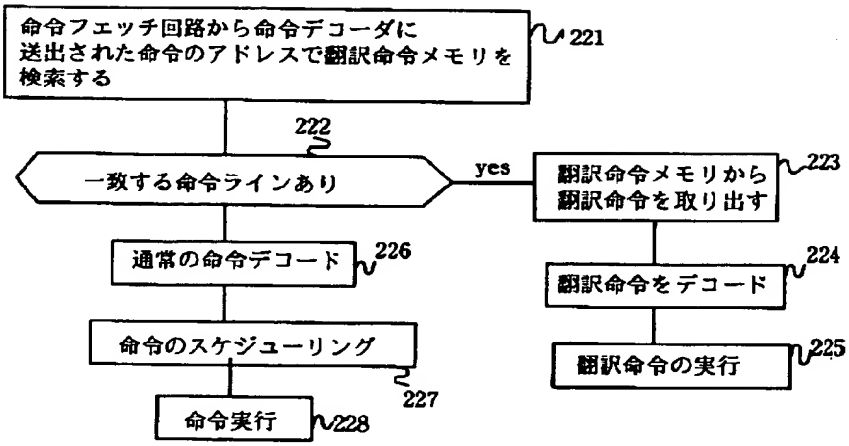
【図４】

図４


0
6
11
16
21
Xフォーマット
OPCD
RT
RA
RB
XO
31
Dフォーマット
OPCD
RT
RA
D

【図２】

図２



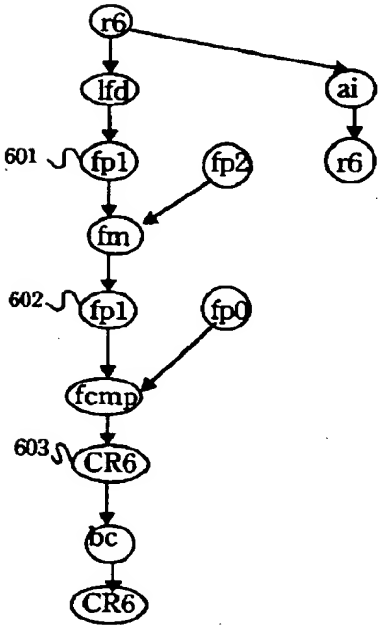
【図５】

図５

```
_L10:   lfd    fp1 8(r6)   ～501
        ai     r6, r6, 4   ～502
        fm     fp1, fp2, fp1  ～503
        fcmp   cr6, fp0, fp1  ～504
        bc     IF,CR6_LT, _L10  ～505
```

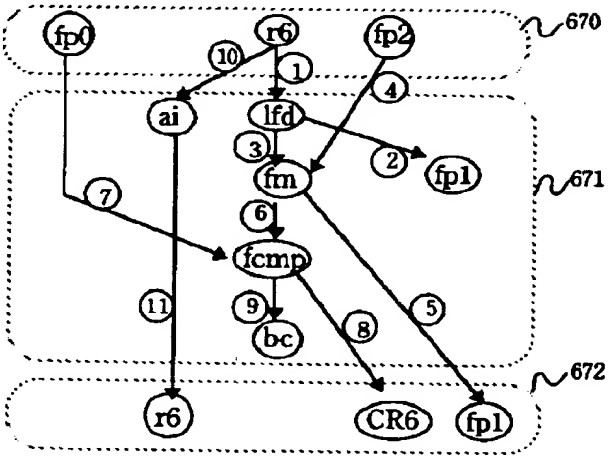
【図６】

図６



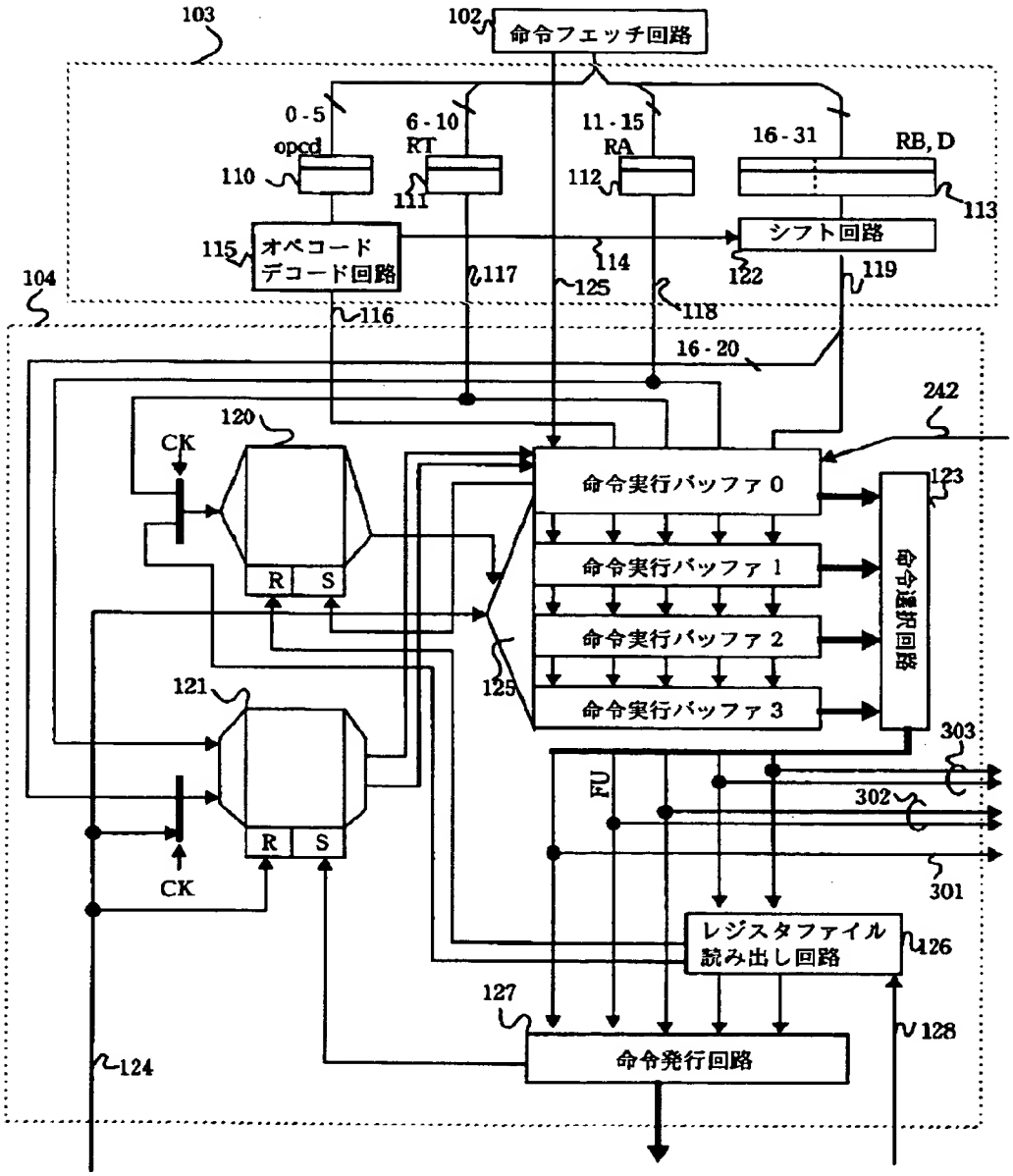
【図７】

図７



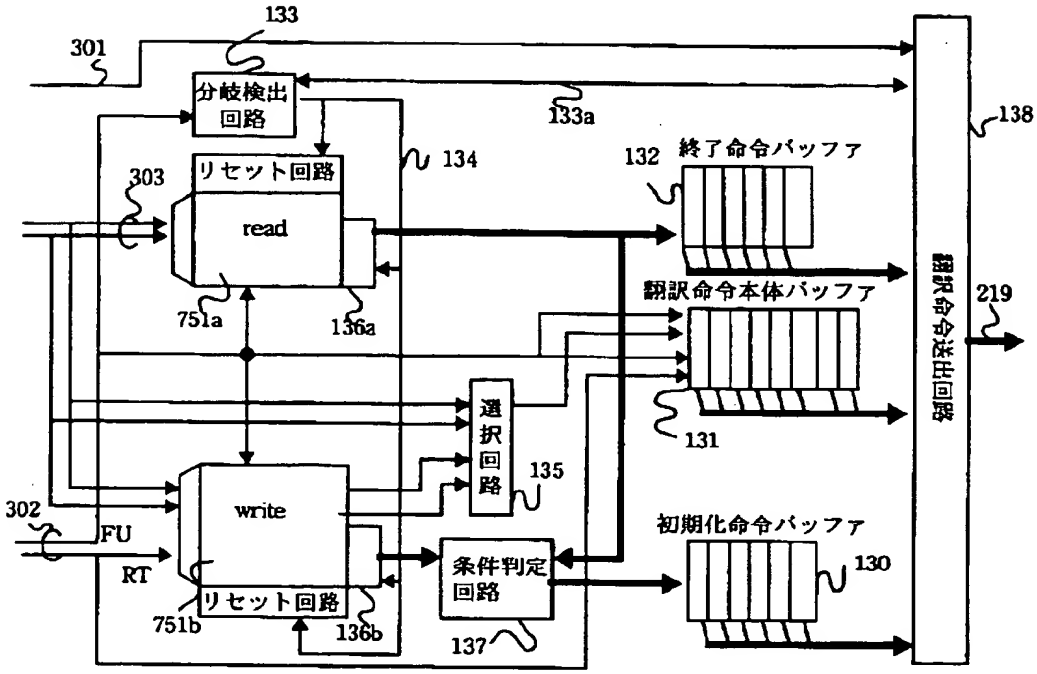
【図３】

図３


103
102
命令フェッチ回路
0-5
opcd
6-10
RT
11-15
RA
16-31
RB, D
110
111
112
113
115
オペコード
デコード回路
シフト回路
104
117
114
125
122
119
116
118
16-20
242
120
CK
命令実行バッファ０
123
R
S
命令実行バッファ１
命令選択回路
命令実行バッファ２
121
125
命令実行バッファ３
303
FU
302
301
CK
レジスタファイル
読み出し回路
126
127
128
124
命令発行回路

【図８】

図８


133
301
分岐検出回路
133a
138
134
132 終了命令バッファ
303
リセット回路
read
翻訳命令本体バッファ
219
翻訳命令送出回路
751a
136a
選択回路
135
131
302
FU
write
RT
初期化命令バッファ
条件判定回路
130
リセット回路
751b
136b
137

【図１０】

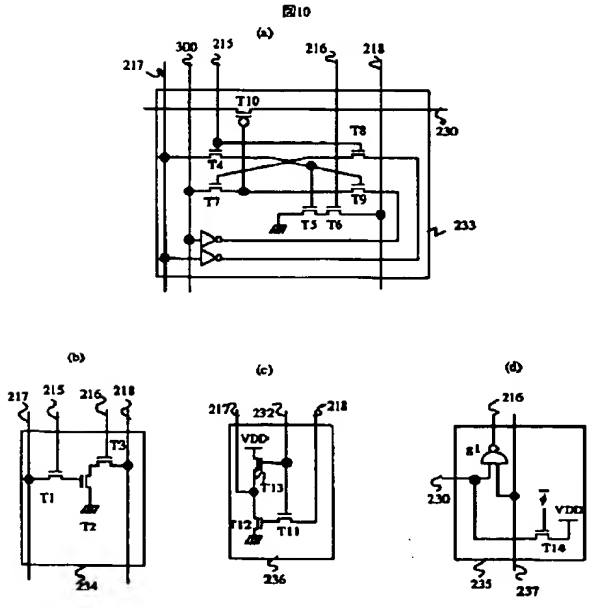


図10
(a)
(b)
(c)
(d)
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
VDD

【図１７】

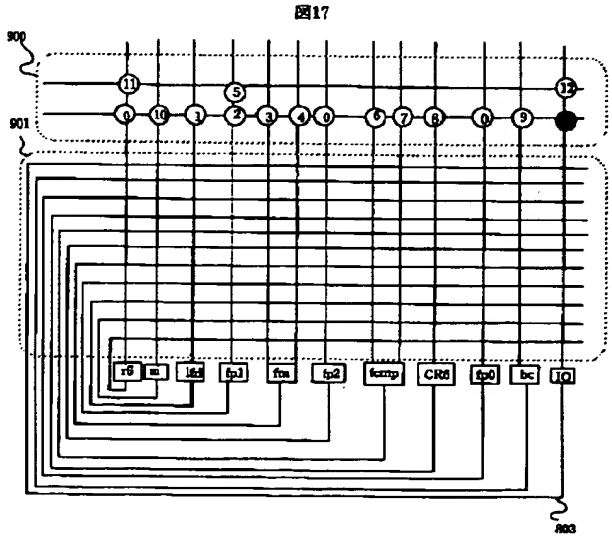
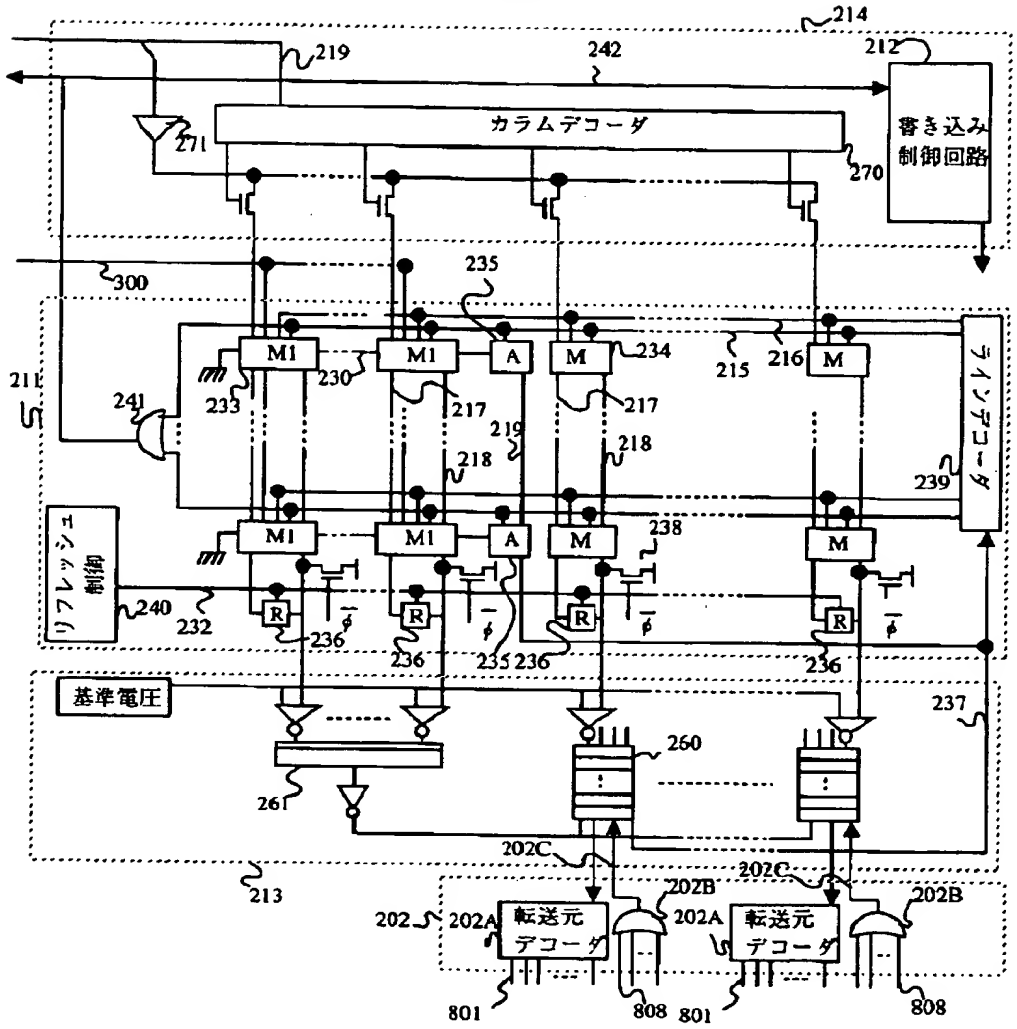


図17
900
901
803

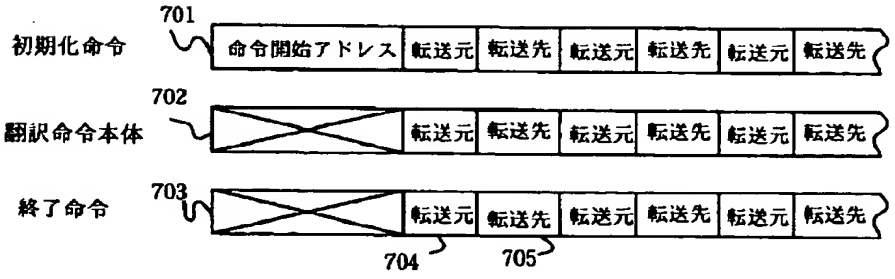
【図９】

図９


214
219
242
212
カラムデコーダ
書き込み制御回路
271
270
300
235
M1
M1
A
M
234
215
216
M
211
241
233
230
217
217
219
218
218
239
ライトコーダ
リフレッシュ制御
M1
M1
A
M
238
M
240
232
R
R
R
R
236
236
235
236
236
基準電圧
237
260
261
202C
202C
213
202
202A
転送元デコーダ
202B
202A
転送元デコーダ
202B
801
808
801
808

【図１１】

図11


701
初期化命令
命令開始アドレス
転送元
転送先
転送元
転送先
転送元
転送先
702
翻訳命令本体
転送元
転送先
転送元
転送先
転送元
転送先
703
終了命令
転送元
転送先
転送元
転送先
転送元
転送先
704
705

【図１２】

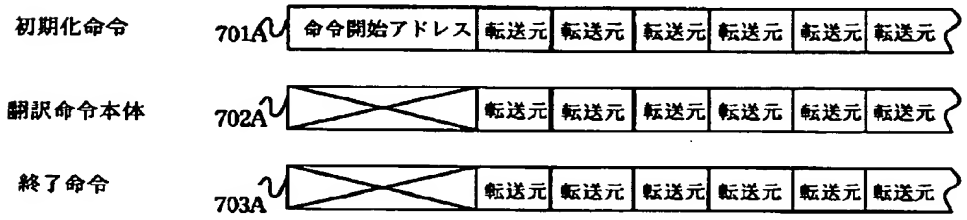
図12

751

	r6		fp1		fp2		fp0		CR6	
	read	write	read	write	read	write	read	write	read	write
lfd fp1 8(r6)	lfd	0	0	lfd	0	0	0	0	0	0
ai r6, r6, 4	ai	ai	0	0	0	0	0	0	0	0
fm fp1, fp2, fp1	0	0	fm	fm	fm	0	0	0	0	0
fcmp 6, fp0, fp1	0	0	fcmp	0	0	0	fcmp	0	0	fcmp
bc IF,CR6_LT, _L10	0	0	0	0	0	0	0	0	bc	0

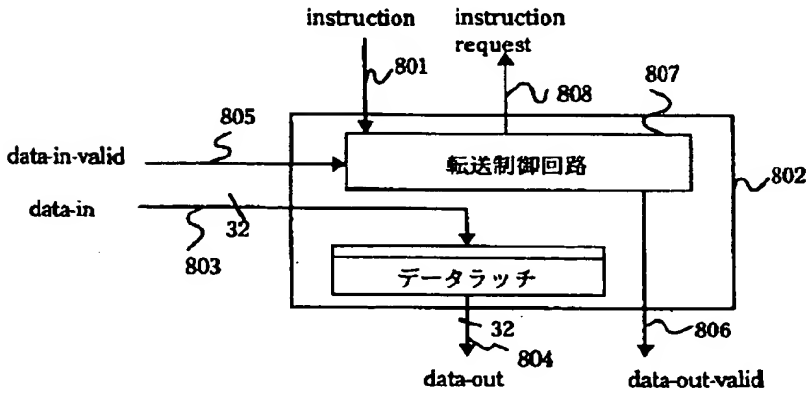
【図１３】

図13



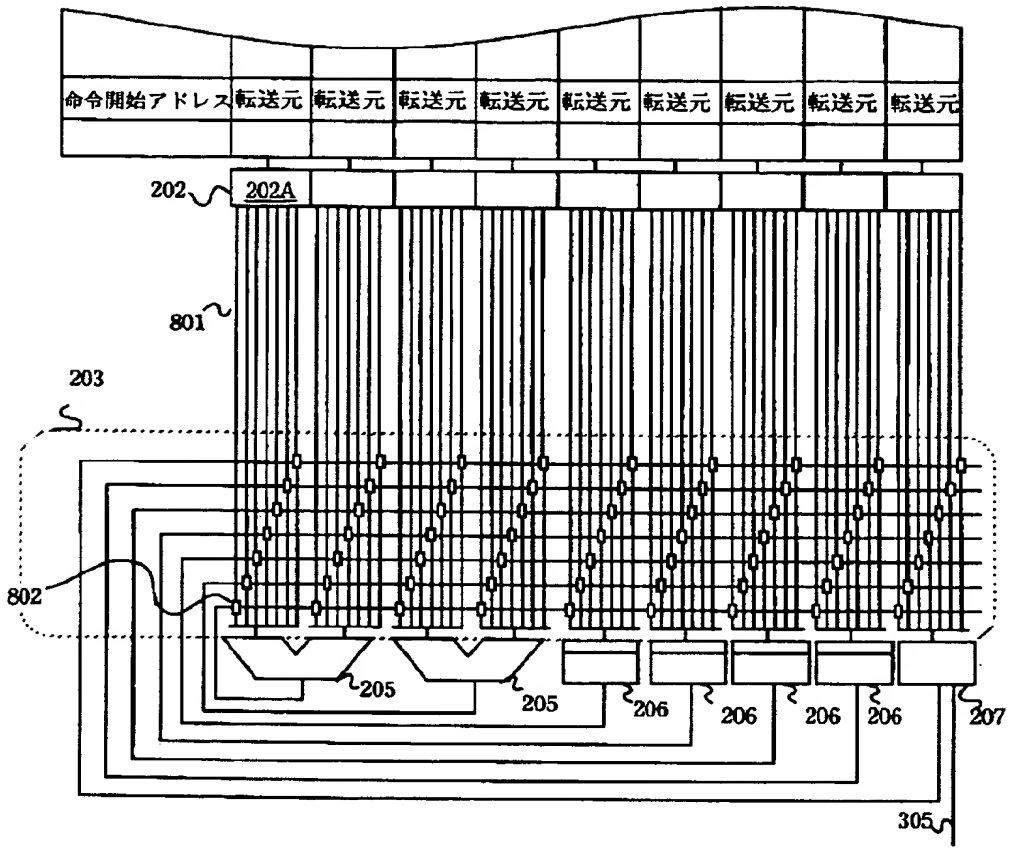
【図１５】

図15



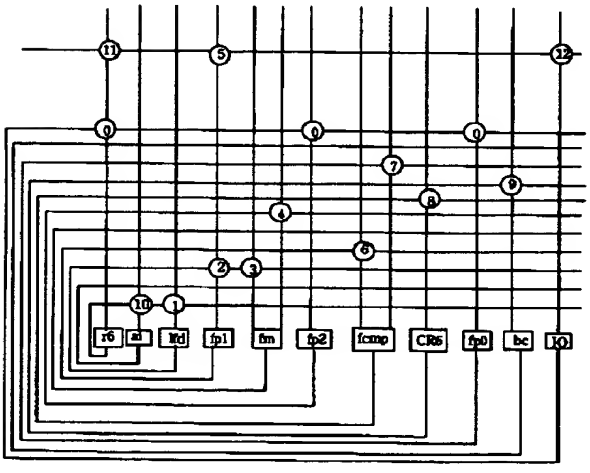
【図１４】

図14



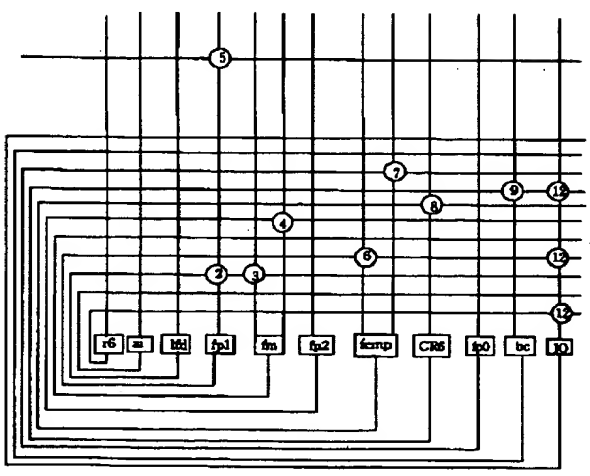
【図１８】

図18



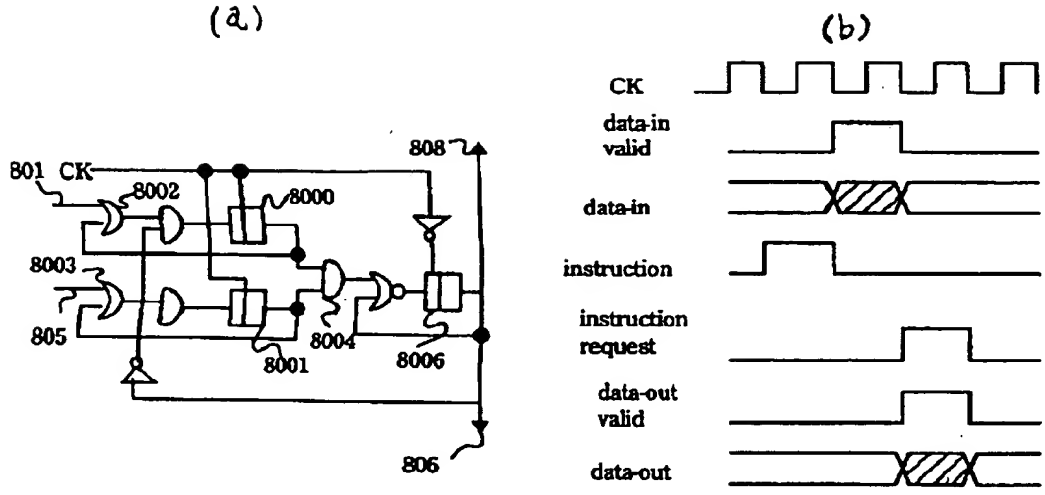
【図２０】

図20



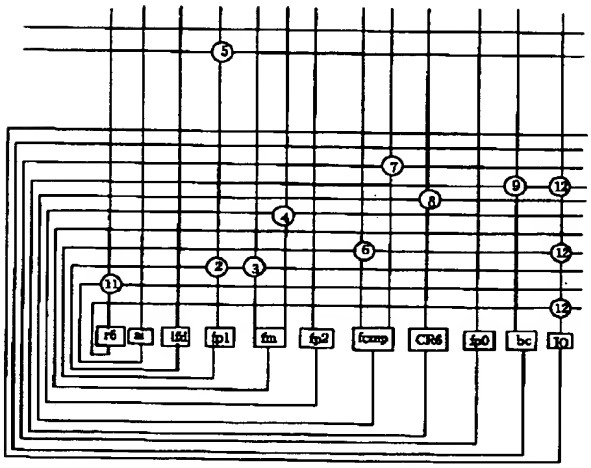
【図１６】

図16



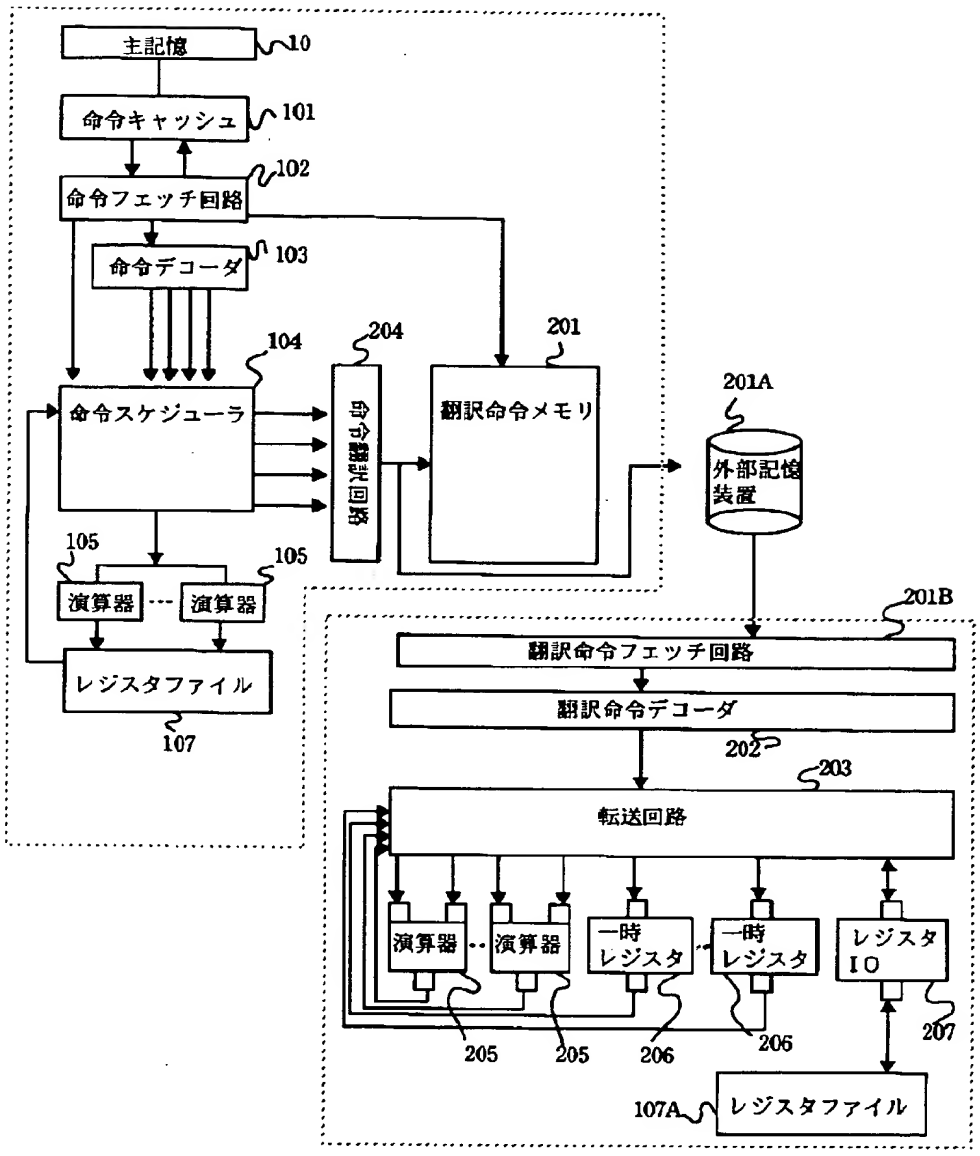
【図１９】

図19



【図２１】

図21


主記憶
10
命令キャッシュ
101
命令フェッチ回路
102
命令デコーダ
103
104
204
201
命令スケジューラ
命令翻訳回路
翻訳命令メモリ
201A
外部記憶装置
105
105
演算器
演算器
レジスタファイル
107
201B
翻訳命令フェッチ回路
翻訳命令デコーダ
202
203
転送回路
演算器
演算器
一時レジスタ
一時レジスタ
レジスタIO
205
205
206
206
207
107A
レジスタファイル